ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

INGENIERO INDUSTRIAL

INTEGRCIÓN DE ENERGÍAS RENOVABLES Y SISTEMAS DE ALMACENAMIENTO EN UNA CASA INTELIGENTE

Autor: Juan Ignacio Ortueta Olartecoechea
Director: Álvaro Sanchez Miralles
Director: Jaime Boal Martín-Larrauri
Director: Christian Calvillo Muñoz

Madrid
Junio 2014

PONTIFICIA UNIVERSITAS COMILLENSIS
MATRITI

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN ACCESO *ABIERTO* (*RESTRINGIDO*) DE DOCUMENTACIÓN

1º. Declaración de la autoría y acreditación de la misma.

El autor D. _____ , como _____ de la UNIVERSIDAD PONTIFICIA COMILLAS (COMILLAS), **DECLARA**

que es el titular de los derechos de propiedad intelectual, objeto de la presente cesión, en relación con la obra_____ _____[1], que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual como titular único o cotitular de la obra.

En caso de ser cotitular, el autor (firmante) declara asimismo que cuenta con el consentimiento de los restantes titulares para hacer la presente cesión. En caso de previa cesión a terceros de derechos de explotación de la obra, el autor declara que tiene la oportuna autorización de dichos titulares de derechos a los fines de esta cesión o bien que retiene la facultad de ceder estos derechos en la forma prevista en la presente cesión y así lo acredita.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad y hacer posible su utilización de *forma libre y gratuita* (*con las limitaciones que más adelante se detallan*) por todos los usuarios del repositorio y del portal e-ciencia, el autor **CEDE** a la Universidad Pontificia Comillas de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución, de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra (a) del apartado siguiente.

3º. Condiciones de la cesión.

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia, el repositorio institucional podrá:

[1] Especificar si es una tesis doctoral, proyecto fin de carrera, proyecto fin de Máster o cualquier otro trabajo que deba ser objeto de evaluación académica

(a) Transformarla para adaptarla a cualquier tecnología susceptible de incorporarla a internet; realizar adaptaciones para hacer posible la utilización de la obra en formatos electrónicos, así como incorporar metadatos para realizar el registro de la obra e incorporar "marcas de agua" o cualquier otro sistema de seguridad o de protección.

(b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato. .

(c) Comunicarla y ponerla a disposición del público a través de un archivo abierto institucional, accesible de modo libre y gratuito a través de internet.[2]

(d) Distribuir copias electrónicas de la obra a los usuarios en un soporte digital. [3]

4º. Derechos del autor.

El autor, en tanto que titular de una obra que cede con carácter no exclusivo a la Universidad por medio de su registro en el Repositorio Institucional tiene derecho a:

a) A que la Universidad identifique claramente su nombre como el autor o propietario de los derechos del documento.

b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.

c) Solicitar la retirada de la obra del repositorio por causa justificada. A tal fin deberá ponerse en contacto con el vicerrector/a de investigación (curiarte@rec.upcomillas.es).

d) Autorizar expresamente a COMILLAS para, en su caso, realizar los trámites necesarios para la obtención del ISBN.

---

[2] En el supuesto de que el autor opte por el acceso restringido, este apartado quedaría redactado en los siguientes términos:

(c) Comunicarla y ponerla a disposición del público a través de un archivo institucional, accesible de modo restringido, en los términos previstos en el Reglamento del Repositorio Institucional

[3] En el supuesto de que el autor opte por el acceso restringido, este apartado quedaría eliminado.

d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.

b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.

c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.

d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

a) Deberes del repositorio Institucional:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.

- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.

- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.

b) Derechos que se reserva el Repositorio institucional respecto de las obras en él registradas:

- retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a …….. de …….. de …….

ACEPTA

Fdo……

Proyecto realizado por el alumno/a:

Juan Ignacio Ortueta Olartecoechea

Fdo:…….. Fecha: …. /…. /….

Autoriza la entrega del proyecto cuya información no es de carácter confidencial

EL DIRECTOR DEL PROYECTO

Dr. Álvaro Sánchez Miralles

Fdo:…….. Fecha: …. /…. /….

EL DIRECTOR DEL PROYECTO

Jaime Boal Martín-Larrauri

Fdo:…….. Fecha: …. /…. /….

EL DIRECTOR DEL PROYECTO

Christian Calvillo Muñoz

Fdo:…….. Fecha: …. /…. /….

Vº Bº DEL COORDINADOR DE PROYECTOS

Dr. Álvaro Sánchez Miralles

Fdo:…….. Fecha: …. /…. /….

Índice de documentos

DOCUMENTO I. MEMORIA

I. Memoria	pág 23 a 96	73 páginas
II. Estudio Económico	pág 97 a 100	4 páginas
III. Manual de usuario	pág 101 a 103	3 páginas
IV. Código Fuente	pág 104 a 144	40 páginas

DOCUMENTO II. PRESUPUESTO

1.	Mediciones	pág 145 a 152	8 páginas
2.	Precios unitarios	pág 153 a 157	5 páginas
3.	Sumas parciales	pág 158 a 163	6 páginas
4.	Presupuesto general	pág 164	1 página

INTEGRACIÓN DE ENERGÍAS RENOVABLES Y SISTEMAS DE ALMACENAMIENTO EN UNA CASA INTELIGENTE

Autor: Ortueta Olartecoechea, Juan Ignacio.

Director: Sánchez Miralles, Álvaro.

Director: Boal Martín-Larrauri, Jaime.

Director: Calvillo Muñoz, Christian Francisco.

Entidad colaboradora: IIT – Universidad Pontificia de Comillas.

RESUMEN DEL PROYECTO

Introducción

El sistema eléctrico del siglo XX se basaba en varios puntos de generación, líneas de transporte y los puntos de consumo. En el siglo XXI surge la idea de la generación distribuida y las microrredes. Esta idea se basa en utilizar redes con pequeños generadores y sistemas de almacenamiento, véase la figura 1. La generación se realiza en el lugar de consumo, por tanto se evitan las pérdidas en el transporte de la energía. Estas microrredes pueden tener conexión a la red de distribución de energía eléctrica o no tenerla.

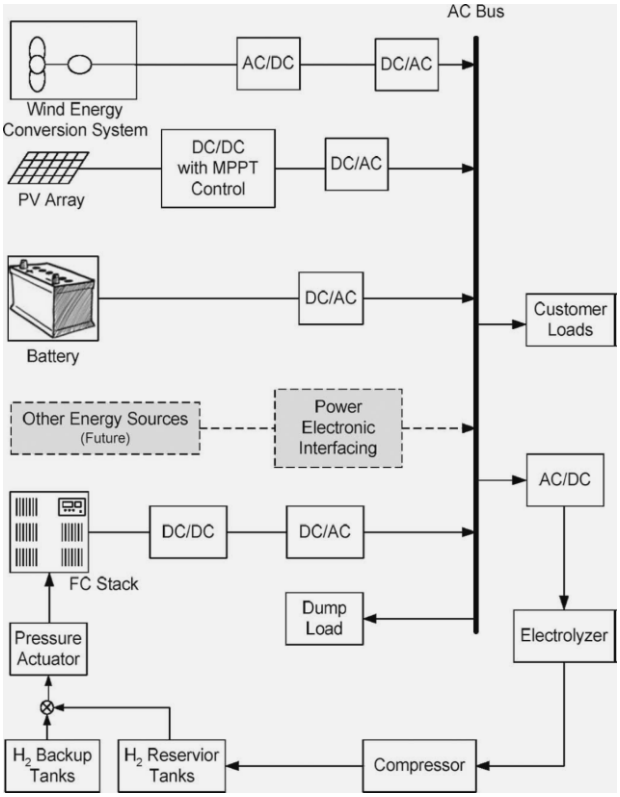

Figura 1. Esquema de la microrred de [1].

La producción de energía en este tipo de redes apuesta, en la mayoría de los casos, por una combinación de diferentes fuentes de energía. En general los diseños incluyen energía solar fotovoltaica, energía eólica y otra fuente distinta. Entre las opciones más usadas se encuentran los generadores diesel, las pilas de combustible o la cogeneración. Actualmente la mayoría de las redes inteligentes de este tipo incluyen un sistema de almacenamiento, mediante baterías o mediante un tanque de hidrógeno para la pila de combustible.

Una de las dificultades que presentan este tipo de redes es la necesidad de tener zonas de la instalación funcionando con tensión alterna y otras zonas funcionando con tensión continua. En general las cargas de los consumidores requieren tensión alterna, mientras que los sistemas de almacenamiento y ciertos generadores, como los paneles solares, emplean tensión continua. Para poder conectar unas zonas con otras es necesario emplear diversos tipos de convertidores: DC/DC, AC/DC y DC/AC. De entre todos ellos juega un papel clave el convertidor DC/AC, también llamado inversor, ya que debe estar sincronizado con la tensión de la red para poder inyectar potencia en la misma. Además cuando la microrred trabaje de forma aislada uno de los convertidores DC/AC será el encargado de fijar el nivel de tensión y la frecuencia de esta.

Al necesitar varios convertidores DC/AC es interesante que estos sean reprogramables. Esto también es útil si la microrred trabaja aislada de la red, ya que así se podría cambiar la tensión o la frecuencia de trabajo. Desgraciadamente la mayoría de los inversores actuales no son reprogramables y si lo son es a un alto precio.

El objetivo del proyecto es desarrollar un inversor, o conversor DC/AC, que posea las siguientes características: una potencia de salida de 200 W, diseño descompuesto en varios módulos, coste inferior a 2000 €, reprogramable y capacidad para funcionar tanto conectado como desconectado de la red de distribución eléctrica.

Arquitectura

Se decidió realizar el proyecto a partir de un diseño ya existente para poder abarcar el proyecto en un año. Tras un análisis de las características de varios inversores, se decidió desarrollar el inversor a partir del inversor AN1444 de Microchip. Los principales motivos fueron: tiene una potencia de salida muy similar a la de los objetivos, optimiza la tensión de funcionamiento maximizando la potencia entregada y el diseño se puede descargar de manera gratuita de una página web de Microchip.

En primer lugar se construyó una réplica de este inversor mediante los planos y lista de componentes proporcionados por Microchip. La arquitectura básica de este inversor está compuesta por varios bloques: un grupo de condensadores de desacoplamiento, un convertidor DC/DC tipo Flyback, un inversor Full Bridge, un filtro EMI y un sistema de control. El software de control de este inversor no permite trabajar sin una tensión de referencia, por lo tanto no es posible trabajar desconectado de una red de distribución eléctrica.

Para solucionar este problema se desarrolló, durante la ejecución del proyecto, un software de control capaz de realizar las tareas necesarias para poder funcionar tanto conectado a red como de manera aislada.

A continuación se comentará el funcionamiento de los controles que se llevan a cabo durante el funcionamiento cuando el microinversor está conectado a la red de distribución, véase la figura 2. Se puede observar que hay tres controles superpuestos: el algoritmo MPPT, el control del balance de carga y el control del Full Bridge.

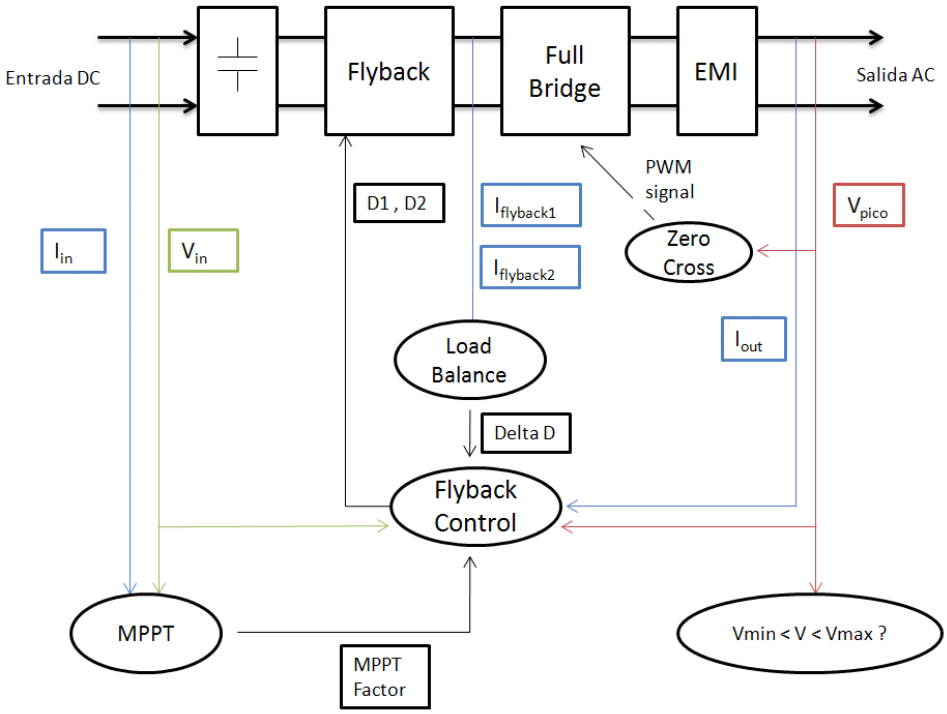

Figura 2. Diagrama de bloques del funcionamiento conectado a red.

El algoritmo MPPT (Maximum Power Piont Tracker) es el encargado de controlar que la zona de tensión continua trabaje a la tensión óptima. Es decir, el valor de tensión a la entrada que maximiza la potencia entregada a la red eléctrica. La función del MPPT, teniendo en cuenta el nivel de tensión a la entrada y la corriente de entrada, genera un coeficiente que se envía a la función de control de los Flyback. Utiliza un algoritmo de perturbación y observación.

Este lazo de control se ve afectado por el lazo que controla el reparto equitativo de la carga entre los dos convertidores Flyback. Este segundo control mide la corriente de cada uno de ellos y calcula una variación del ciclo de trabajo (duty cycle). Es un control de tipo proporcional integral. Esta variación del ciclo de trabajo se suma al ciclo de trabajo de uno de los Flyback y se resta al ciclo de trabajo del otro Flyback.

Para unir estos dos controles está la función Flyback Control, que posee un control proporcional integral. Los datos de entrada de dicha función son: la tensión máxima a la salida, la corriente de salida, la tensión de entrada, el factor del algoritmo MPPT y la

variación de ciclos de trabajo del balance de carga. Con esta información la función genera los ciclos de trabajo de cada uno de los Flyback, estos valores digitales se comunican a los transistores de control mediante el PWM del microprocesador.

Por otro lado está el control del inversor Full Bridge, este control es el encargado de controlar el sincronismo entre la tensión de referencia y la tensión generada. Para ello toma los valores de pico y los cruces por cero de la tensión de referencia. Verifica que el pico de la onda generada coincida con el pico de la tensión de red y en caso contrario calcula las modificaciones necesarias. Este lazo de control actúa sobre el inversor Full Bridge mediante otro de los generadores PWM que posee el microprocesador.

Cuando el inversor trabaja de manera aislada se producen algunos cambios en los controles, véase la figura 3. Se calcula una referencia teórica mediante un temporizador que salta cada 100 ms y una función seno tabulada. La función que controla el inversor Full Bridge en vez de seguir la referencia marcada por la tensión de salida sigue la referencia teórica. Igualmente la función que calcula el factor de servicio de los Flyback calcula dicho factor teniendo en cuenta el nivel de tensión de la referencia teórica calculada.

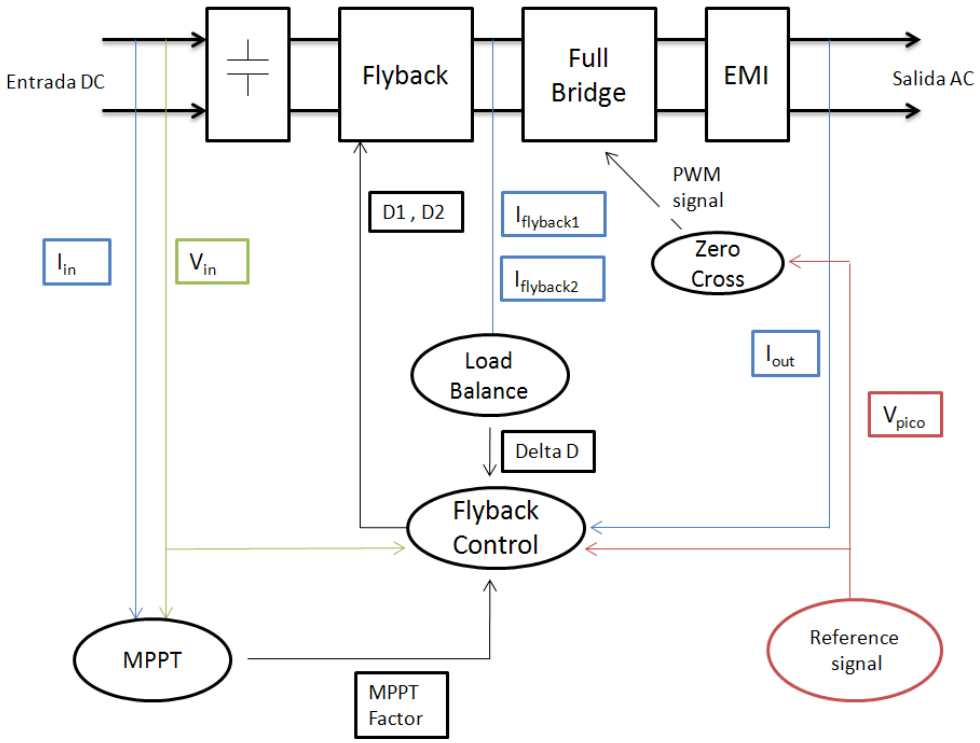

Figura 3. Diagrama de bloques del funcionamiento en modo isla.

Resultados

Se realizaron dos pruebas: la verificación del funcionamiento conectado a la red de distribución y la verificación del funcionamiento en modo isla. En la primera de ellas se observó el adecuado funcionamiento del microinversor cuando ha de seguir la referencia de tensión marcada por la red de distribución eléctrica, comprobándose que la

referencia se siguió de manera correcta. En la segunda prueba realizada en modo isla se observó el correcto funcionamiento del microinversor cuando ha de generar la referencia de tensión. Esta referencia se generó de manera adecuada y la distorsión total armónica obtenida fue menor del 5%.

Los resultados obtenidos fueron: diseños en Eagle de cada uno de los módulos, fabricación del inversor con un coste de 887.6 €, se logró el correcto funcionamiento del microinversor conectado a la red de distribución y se alcanzó el objetivo de que el microinversor funcionase en modo isla.

Conclusiones

Como conclusión se ha conseguido desarrollar un inversor capaz de funcionar conectado a la red y en modo isla, con un coste inferior al presupuestado, por lo que se ha cumplido el objetivo principal del proyecto.

Este proyecto ha desarrollado una base sólida para que, en un futuro cercano, se consiga desarrollar una microrred completa y eficiente. Mediante el desarrollo de las microrredes se conseguirá una reducción del consumo eléctrico, debido a su fácil integración con energías renovables. Además al promover la generación distribuida se reducirán las pérdidas en las redes de transporte. Esta reducción del consumo y de las pérdidas de energía permitirá el desarrollo de un futuro más sostenible y ecológico para todos.

Referencias

[1] C. Wang and M. H. Nehrir, “Power Management of a Stand-Alone Wind/Photovoltaic/Fuel Cell Energy System,” IEEE Transactions on Energy Conversion, vol. 23, no. 3, pp. 957–967, 2008.

ABSTRACT

Introduction

The electrical system of the twentieth century was based on several points of generation, transmission lines and points of consumption. In the twenty first century the idea of distributed generation and microgrids comes out. This idea is based on using networks with small generators and storage systems. An example is shown in Figure 1. Generation is performed at the place of consumption, thus avoiding losses in the transport of energy. These microgrids can have a connection to the electricity distribution network or not.

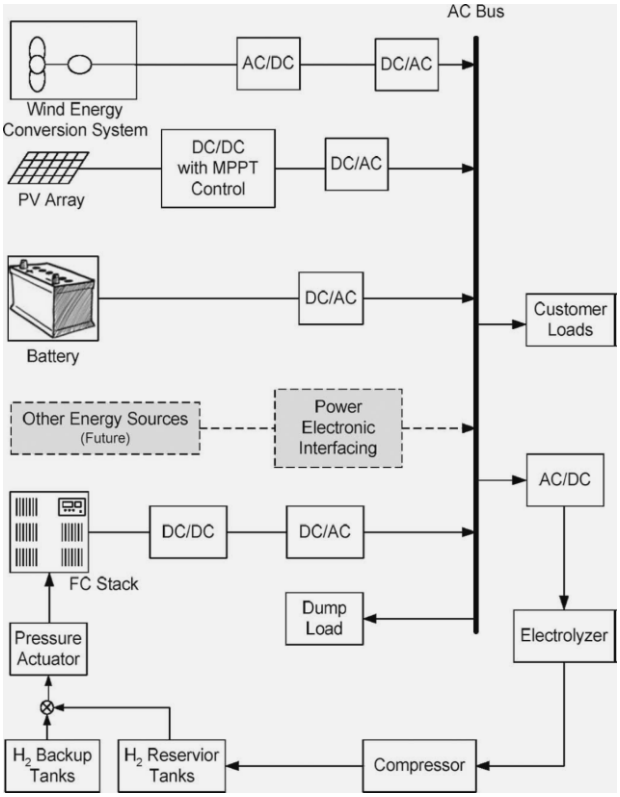

Figure 1. Schematic of the microgrid from [1].

In most cases energy production in this type of network is committed to a combination of different energy sources. In general designs include solar energy sources, wind energy sources and another different energy source. Among the most commonly used options there are diesel generators, fuel cells and cogeneration. Currently most of this type of grids includes a storage system by batteries or by hydrogen tank. The hydrogen stored in the tank can be transformed to electricity by fuel cells.

One possible problem in this kind of networks is the need of having some areas of the facility operating on AC voltage and other areas operating on DC voltage. In general consumer loads require AC voltage, whereas storage systems and certain generators like

solar panels need DC voltage. In order to connect AC areas to DC areas, there are needed different types of converters, such as: DC/DC converters, AC/DC converters and DC/AC converters. Among all of them, the DC/AC converter plays a key role, since it must be synchronized with the grid voltage to inject power in it. The DC/AC converter can be called inverter too. Also when the microgrid works in isolated operation mode one of the DC/AC converters will be responsible for setting the voltage level and grid´s frequency.

As several DC/AC converters are needed, it is important that these are reprogrammable. This characteristic is also useful if the microgird is operating in isolated mode, as it will be a way of changing grid´s frequency and voltage level. Unfortunately most current inverters are not reprogrammable and if this happens it involves a very high cost.

This project aims to develop an inverter or DC/AC converter, that has the following characteristics: an output power of 200 W, a design decomposed into several modules, manufacture cost under 2000 €, reprogrammable and the ability to work both connected to or disconnected from the electricity distribution network.

Architecture

It was decided to carry out the project from an existing design to conclude it in only one year. After analysing characteristics of several inverters, it was decided to develop the inverter based on AN1444 Microchip inverter. The main reasons were: it has an output power which is very similar to goal´s output power, it optimizes the input voltage to maximize output power and the design can be downloaded for free from Microchip´s web page.

First of all, an AN1444 inverter reproduction was constructed using Microchip´s plans and Microchip´s bill of materials. The basic architecture of this inverter is composed of several blocks: a group of decoupling capacitors, a DC/DC Flyback converter, a Full Bridge inverter, an EMI filter and a control system. The control software does not allow the inverter to operate without a reference voltage, so it is not possible to work disconnected from a power distribution network.

In order to solve this problem, a control software was developed during project implementation. This software is able to perform every task needed to operate connected to and disconnected from power distribution network.

The control tasks carried out during grid connected mode will be discussed. A block diagram of grid connected operation is shown in figure 2. There are three control loops at the same time: MPPT algorithm, load balance control and Full Bridge control.

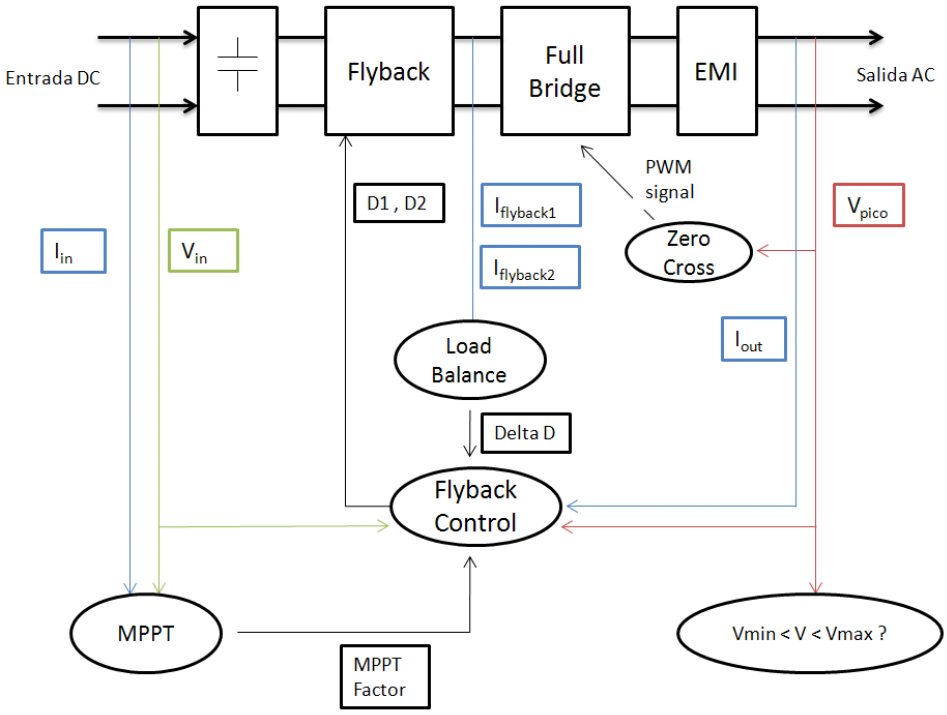

Figure 2.Block diagram of grid connected operation.

MPPT algorithm (Maximum Power Point Tracker) is in charge of detecting the optimal input voltage, which is the input voltage that maximizes the output power. MPPT function takes into account input voltage and input current, then it generates a factor that is sent to Flyback control function. MPPT function operates with perturb and observe method.

This control loop is affected by load balance control, its main task consist in verifying that load is distributed equally between the two Flyback converters. This second control measures both Flyback currents and computes the duty cycle´s variation needed. It is a proportional integral control. The variation of duty cycles is added to the duty cycle of one Flyback and subtracted to the other.

The Flyback control function links these two control loops, it is a proportional integer control. This function has several input data: output peak voltage, output current, input voltage, MPPT factor and duty cycle variation. After gathering this information, this function generates the duty cycle of both Flyback converters. Then these digital values are transferred to the Flyback converters through PWM signals. These signals are sent to the transistors of each Flyback converter.

Regarding to the Full Bridge control, it is in charge of synchronizing the generated voltage with the reference voltage. This function has two input data from reference voltage: time of peak voltage (negative and positive) and time of zero cross. It verifies whether the reference´s peak voltage is generated at the same time as the peak voltage or not. If peak voltages do not happen at the same time, this control modifies the generated voltage through a PWM signal.

When the inverter operates in isolated mode, some changes are made to the control loops. These changes are shown in figure 3. A theoretical reference is calculated every 100 ms using a timer and a tabulated sine wave. The function that controls Full Bridge inverter follows the theoretical voltage reference instead of following the output voltage reference. Also the Flyback control function, which calculates the Flybacks´ duty cycle, takes into account the theoretical voltage reference instead of the output voltage reference.

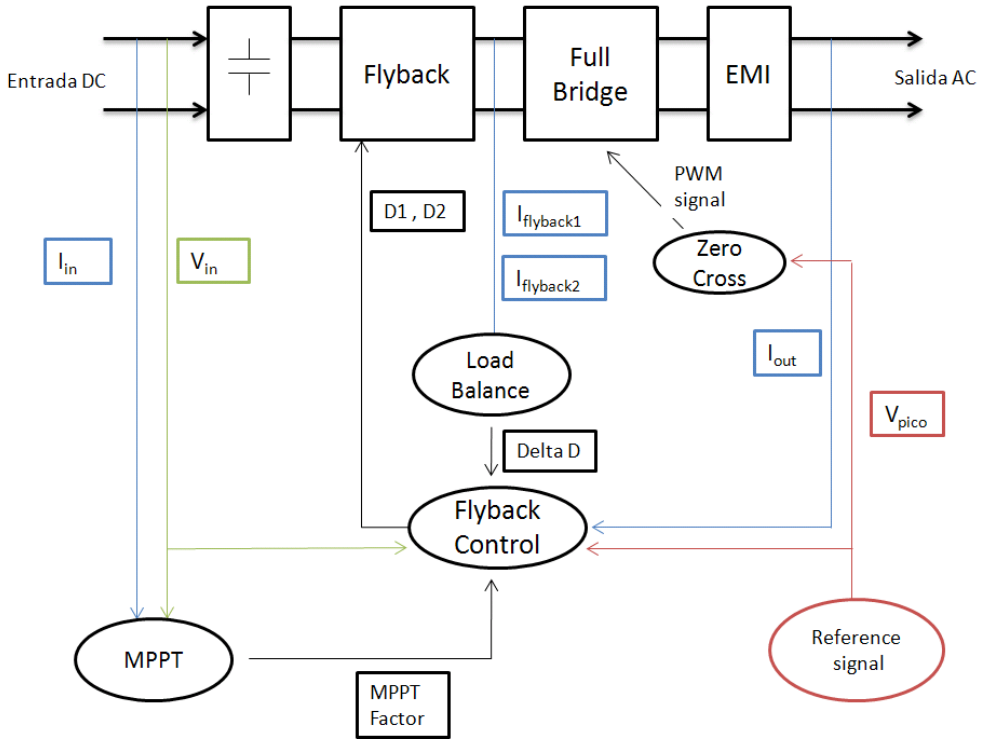

Figure 3.Block diagram of isolated mode operation.

Results

It were tested both the connected operation mode and the isolated operation mode. In the first one the proper microinverter operation was observed when the electricity distribution network must be followed. In the second one the proper operation was observed when the microinverter must generate his reference voltage. This reference was generated appropriately and the total harmonic distortion obtained was under 5%.

The results obtained were: Eagle designs of each module, inverter manufacture with a cost of 887.6 €, correct microinverter operation in connected mode and in isolated mode.

Conclusions

In conclusion it has been possible to develop an inverter that is able to work connected to and disconnected from the electricity distribution network. As its cost was lower than budgeted cost, the main objective of this project was fulfilled.

This project has developed the bases to achieve, in a near future, a complete and efficient microgrid development. Through the development of microgrids electricity consumption will be reduced, due to its easy integration with renewable energies. In addition by promoting distributed generation, losses in transmission networks will be minimized. This reduction in energy consumption and energy losses will create a more sustainable and ecological future.

References

[1] C. Wang and M. H. Nehrir, "Power Management of a Stand-Alone Wind/Photovoltaic/Fuel Cell Energy System," IEEE Transactions on Energy Conversion, vol. 23, no. 3, pp. 957–967, 2008.

Agradecimientos

En primer lugar me gustaría agradecer el apoyo proporcionado por mis padres durante los cinco años de carrera, sobre todo en los momentos más difíciles cuando todo parecía imposible.

Quisiera agradecer a Álvaro Sánchez, uno de mis directores de proyecto, sus indicaciones cruciales para guiar el desarrollo de mi proyecto fin de carrera.

Desearía agradecer también a Jaime Boal, otro de mis directores quien estuvo presente cada día, por su paciencia a la hora de enseñarme y por su ayuda en la resolución de los problemas.

Quisiera agradecer a Christian Calvillo, el último de mis directores, por transmitirme tranquilidad incluso en los momentos más difíciles y por sus excelentes explicaciones.

No quisiera olvidarme de agradecer al personal de mantenimiento del departamento de electrónica, especialmente a José María Bautista, que hicieron lo imposible para reparar los circuitos tras los diversos accidentes.

Por último me gustaría agradecer a mis compañeros y amigos de la universidad, por animarme siempre que lo necesité y hacerme reír a pesar de todos los problemas.

DOCUMENTO I

MEMORIA

Índice de la memoria

Parte I Memoria ***30***

Capítulo 1 Introducción ***31***

1.1 Estudio de las tecnologías existentes **31**
1.1.1 Hardware 31
1.1.2 Algoritmos de control 38
1.1.3 Sistemas de comunicación 45

1.2 Motivación del proyecto **50**

1.3 Objetivos **51**

1.4 Metodología **51**

1.5 Recursos y herramientas empleadas **53**

Capítulo 2 Arquitectura ***54***

2.1 Introducción **54**

2.2 Hardware **54**

2.3 Software **55**

2.4 Algoritmos de control **56**

2.5 Algoritmo MPPT **57**

Capítulo 3 Análisis del hardware ***59***

3.1 Introducción **59**

3.2 Sensor de corriente alterna **59**

3.3 Sensor de tensión de entrada **61**

3.4 Sensor de tensión alterna **62**

3.5 Sensor de corriente del Flyback **64**

3.6 Convertidor Flyback **66**

3.7 Controlador del convertidor Flyback **69**

3.8 Puente H **71**

3.9 Controlador del puente H **73**

3.10 Protección de sobretensión del Flyback **75**

3.11 Conexión con el microprocesador 77
3.12 Filtro EMI 82
Capítulo 4 Desarrollo del software 84
4.1 Introducción 84
4.2 Funcionamiento conectado a la red de distribución 84
4.3 Funcionamiento en modo isla 89
Capítulo 5 Resultados 91
Capítulo 6 Conclusiones 93
Capítulo 7 Futuros desarrollos 94
Bibliografía 95
Parte II Estudio económico 97
Capítulo 1 Estudio económico 98
Parte III Manual de usuario 101
Capítulo 1 Manual de usuario 102
Parte IV Código fuente 104
Capítulo 1 Defines.h 106
Capítulo 2 Init.c 110
Capítulo 3 Main.c 115
Capítulo 4 Isr.c 116
Capítulo 5 Statemachine.c 130

Índice de figuras

Figura 1. Esquema de la microrred de [4]. 35
Figura 2. Estructura de la microrred de [5]. 36
Figura 3. Estructura de la microrred de [6]. 37
Figura 4. Arquitectura del control y sistema de gestión de [10]. 40
Figura 5. Arquitectura del control y sistema de gestión de [12]. 42
Figura 6. Esquema de la microrred de [16]. 44
Figura 7. Esquema de la microrred de [18]. 47
Figura 8. Esquema del sistema de comunicación de la microrred de [19]. 48
Figura 9. Esquema de la microrred de [20]. 49
Figura 10. Niveles de una red domestica [21]. 50
Figura 11. Cronograma del proyecto. 52
Figura 12. Diagrama de bloques del hardware del micronversor AN1444. 55
Figura 13.Funcionamiento del algoritmo MPPT. 58
Figura 14. Esquema del sensor de corriente alterna. 60
Figura 15. Circuito impreso del sensor de corriente alterna. 60
Figura 16. Esquema del sensor de tensión de entrada. 61
Figura 17. Circuito impreso del sensor de tensión de entrada. 62
Figura 18. Esquema del sensor de tensión alterna. 63
Figura 19. Circuito impreso del sensor de tensión alterna. 64
Figura 20. Esquema del sensor de corriente del Flyback. 65
Figura 21. Circuito impreso del sensor de corriente del Flyback. 66
Figura 22. Esquema del convertidor Flyback. 67
Figura 23. Circuito impreso del convertidor Flyback. 69
Figura 24. Esquema del controlador del Flyback. 70
Figura 25. Circuito impreso del controlador del Flyback. 71
Figura 26. Esquema del puente H. 72
Figura 27. Circuito impreso del puente H. 73
Figura 28. Esquema del controlador del puente H. 74

Figura 29. Circuito impreso del controlador del puente H. 75
Figura 30. Esquema del circuito de sobretensión del Flyback. 76
Figura 31. Circuito impreso de protección contra sobretensión en el Flyback. 77
Figura 32. Esquema del circuito de conexión con el microprocesador. 78
Figura 33. Circuito de acondicionamiento para la alimentación a 12 V. 78
Figura 34. Circuito de acondicionamiento para la alimentación a 5 V. 79
Figura 35. Circuito de acondicionamiento para la alimentación a 3.3 V. 79
Figura 36. Circuito de acondicionamiento para los LEDs y el interruptor. 80
Figura 37. Circuito para la comunicación con el PC. 80
Figura 38. Circuito impreso de conexión con el microprocesador. 81
Figura 39. Esquema del filtro EMI. 82
Figura 40. Circuito impreso del filtro EMI. 83
Figura 41. Máquina de estados del código original. 85
Figura 42. Máquina de estados del código desarrollado. 86
Figura 43. Diagrama de bloques del funcionamiento conectado a red. 88
Figura 44. Diagrama de bloques del funcionamiento en modo isla. 90
Figura 45. Rendimientos del microinversor conectado a la red de distribución. .. 91
Figura 46. Ciclo de vida de un producto. 99

Índice de tablas

Tabla 1. Comparativa relacionada con el hardware. 32
Tabla 2. Comparativa relacionada con el control de la microrred. 39
Tabla 3. Prioridades de las cargas y preferencias en [15]. 43
Tabla 4. Comparativa de los sistemas de comunicación. 45
Tabla 5. Errores posibles del microinversor. 103

Parte I MEMORIA

Capítulo 1 INTRODUCCIÓN

En este capítulo se hace una introducción del proyecto. En la sección 1.1 se realiza un análisis del estado del arte. Posteriormente en la sección 1.2 se revela la motivación del proyecto. A continuación se indican: los objetivos del proyecto, en la sección 1.3, y la metodología, en la sección 1.4. Por último se muestran, en la sección 1.5, los recursos y herramientas empleadas.

1.1 ESTUDIO DE LAS TECNOLOGÍAS EXISTENTES

El estudio del estado del arte se dividirá en 3 partes: los artículos relacionados con el hardware de la microrred, los relacionados con el control de la microrred y los relacionados con el sistema de comunicación empleado en la microrred.

1.1.1 HARDWARE

En la Tabla 1 se puede ver una comparación de los diferentes trabajos considerados en este estudio del estado del arte relacionados con el hardware.

Trabajo	*Fuentes de energía*	*Tipo de inversor*	*Almacenamiento*	*Tamaño*
[1]	Solar fotovoltaica y solar térmica	Multilevel inverter	Tanque de agua	42.5 W
[2]	Solar fotovoltaica y eólica	Bidirectional inverter-charger	Baterías 44 KWh	4 KW
[3]	Pila de combustible (H_2) o turbina de gas	Economic bidirectional inverter	Baterías y supercondensadores	-
[4]	Solar fotovoltaica, eólica y pila de combustible (H_2)	-	Baterías 10 KWh y tanque de hidrógeno 18 KW	>100 KW
[5]	-	Back to back converter	-	>1 MW
[6]	Solar fotovoltaica y pila de combustible	Back to back converter	-	>1 MW
[7]	-	Back to back converter	-	120 KW

Tabla 1. Comparativa relacionada con el hardware.

A continuación se comentan detalladamente cada uno de los trabajos analizados.

En [1] se propone una red inteligente de un edificio que se basa en la energía solar fotovoltaica y, al mismo tiempo, en la energía solar térmica. Hay que tener en cuenta que los paneles solares producen electricidad en corriente continua y las cargas de los usuarios necesitan en general corriente alterna.

En este trabajo la prioridad más importante es la calidad del abastecimiento de energía y en este sentido se deben controlar varios factores como: la tensión, la frecuencia de la corriente, la distorsión armónica, el factor de potencia, la potencia reactiva y el tiempo de respuesta del sistema. Para la conversión de corriente continua a alterna se usa un inversor PWM multinivel porque gracias a él se pueden alcanzar niveles altos de tensión con un nivel bajo de armónicos en la tensión.

La segunda prioridad del sistema es minimizar las pérdidas en los procesos de transformación de la energía. Para evitar dichas pérdidas se usa la energía solar térmica para enfriar el ambiente, para calentar el agua o para calefacción. Además, mediante depósitos de agua caliente o de hielo, se puede almacenar dicha energía fácilmente para usarla en otro momento.

En [2] se analiza el funcionamiento de la red inteligente de un hogar en la que se instala una potencia de 4 kW de generación máxima. Se diseñó la red para cubrir las necesidades de un hogar medio de Nueva Inglaterra con una probabilidad de pérdida de abastecimiento de un día cada diez años. En primer lugar se consultó a un centro meteorológico cercano las características del viento y la irradiación solar de la zona. Con estos datos, utilizando distribuciones de probabilidad, se calculó la potencia de energía solar óptima (2.5 kW) y la capacidad de almacenamiento óptima de la batería (44kWh) para minimizar el coste del sistema.

El sistema consta de cuatro partes principales: los paneles solares fotovoltaicos, la turbina eólica que es un generador síncrono de imanes permanentes de 1.5 kW de potencia, las unidades de acondicionamiento de energía y las baterías.

Las unidades de acondicionamiento de energía están compuestas por: un microprocesador que fija la tensión de los paneles solares para maximizar la potencia generada en cada momento, un rectificador que transforma la energía alterna de la turbina en energía continua, un inversor-cargador bidireccional que conecta el sistema con la red y un inversor que conecta el sistema con la carga mediante un transformador.

A continuación se hace un análisis del sistema y se comprueba que el sistema no funciona como se esperaba debido a que las condiciones meteorológicas no son iguales a las del centro meteorológico cercano, por este motivo la generación solar fotovoltaica está en torno al 78% de lo esperado y la eólica en torno al 6% de lo esperado.

Por último, se elabora un estudio económico (año 2001) del sistema en el que se estima que un particular debería pagar $112.88 al mes. Teniendo en cuenta que el precio del kW sería de $0.41 frente a $0.12, precio que se paga a la compañía eléctrica de la zona, se ve que es más caro instalar el sistema que seguir pagando a la compañía eléctrica. No obstante se cree que este precio se irá reduciendo con el tiempo pudiendo incluso llegar a ser competitivo.

En [3] se analiza que para no tener abastecimiento de energía intermitente o variable la solución es añadir un sistema de almacenamiento de energía. La solución propuesta se basa en la topología de multipuerto centralizada, compuesta por un convertidor DC/DC al que se le conectan varios almacenamientos o generación de energía y un inversor bidireccional que conecta el resto de elementos con la red inteligente.

El diseño mostrado está formado por cuatro elementos conectados entre sí mediante un transformador de continua. El primer elemento es una batería conectada al transformador

mediante un convertidor Boost Half Bridge, el segundo elemento es un supercondensador conectado al transformador a través de un convertidor Buck-Boost de tres niveles, el tercer elemento es un inversor que conecta la generación eléctrica de una turbina de gas al transformador y por último, el cuarto elemento es la conexión con la red inteligente mediante un convertidor Half Bridge conectado a un inversor económico. El inversor económico se basa en la sustitución de dos de los interruptores por dos condensadores haciendo al inversor más compacto y económico.

Para el control de la red se usa un sistema de control de dos niveles: el nivel más alto gestiona el flujo de potencia y el nivel más bajo se encarga de controlar los diferentes puertos. El nivel superior activa las baterías cuando la demanda es constante y en los transitorios de potencia activa el supercondensador. El nivel inferior controla el desfase del PWM de cada puerto para obtener la potencia activa y reactiva demandada por la red inteligente.

Se comprueba, mediante simulación del modo isla, que sin almacenamiento hay oscilaciones en la frecuencia de la red y en su voltaje, pero con el almacenamiento diseñado los problemas desaparecen.

En [4] se reflexiona acerca de las fuentes de energía renovables, teniendo cada una de ellas ciertos inconvenientes, y se concluye que lo más apropiado es hacer un sistema híbrido con varias fuentes de energía. Se presenta dicho sistema con paneles solares, una turbina eólica, una pila de combustible de H_2 y baterías. Además el sistema incluye un electrolizador que genera H_2 consumiendo electricidad. La pila de combustible se usa como almacenamiento a largo plazo y las baterías como almacenamiento a corto plazo. Para unir las diferentes fuentes de energía se usan varios convertidores e inversores, véase la Figura 1. *Esquema de la microrred de [4]*.Figura 1.

La turbina genera en alterna, se pasa a corriente continua para después utilizar un inversor que lo conecta a la red de 60 Hz. Los paneles solares y las baterías se conectan a la red mediante otros dos inversores. La pila de combustible se conecta a la misma red a través de un convertidor de continua a continua y un inversor. También se usa un rectificador para alimentar al electrolizador.

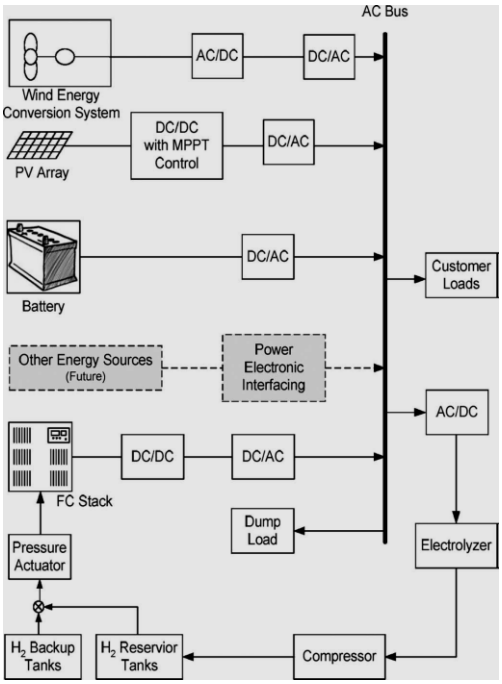

Figura 1. Esquema de la microrred de [4].

Cuando hay un exceso de producción por parte de los paneles solares y la turbina eólica se pone en marcha el electrolizador que genera H_2 y lo introduce en un tanque de reserva. Si, por el contrario, hay un déficit en la producción de energía se activa la pila de combustible que transforma el H_2 almacenado en energía eléctrica. Por último se efectúan varias simulaciones con varios escenarios que muestran la factibilidad y eficacia del sistema híbrido propuesto.

En [5] se indica que para obtener un mejor rendimiento de los generadores y una gestión energética más eficiente es importante alcanzar un control del flujo de potencia entre la red de distribución y la microrred.

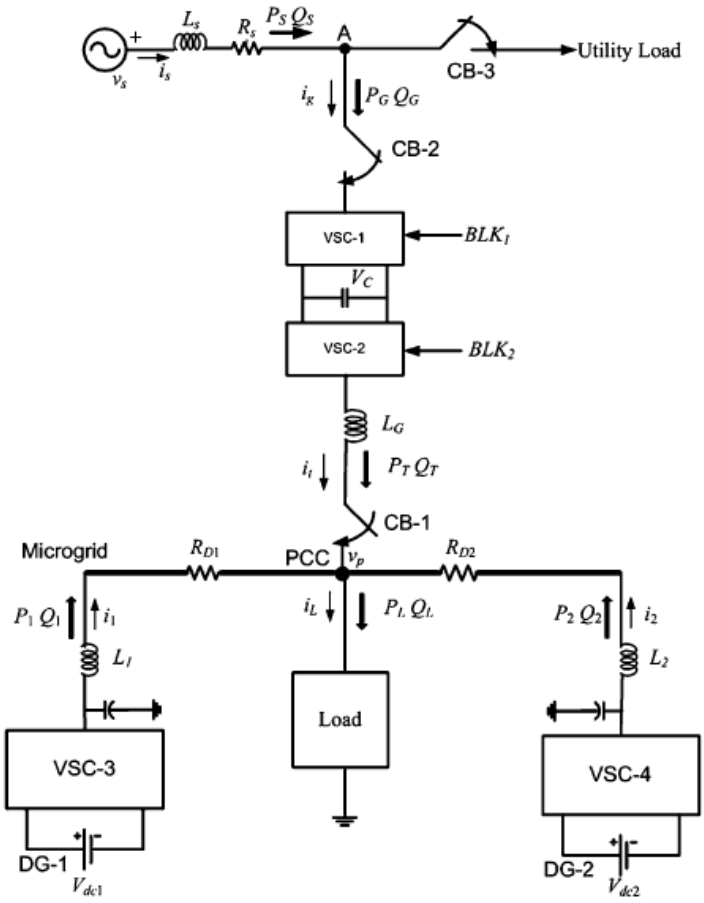

Figura 2. Estructura de la microrred de [5].

Se presenta un esquema de microrred basado en el convertidor back to back. El flujo bidireccional de potencia se obtiene controlando cada uno de los dos convertidores que contiene el convertidor back to back, cuando uno funciona como inversor el otro funciona como rectificador y al revés.

Además el convertidor proporciona aislamiento a la frecuencia y al voltaje de la microrred con respecto a la frecuencia y al voltaje de la red de distribución. El esquema del convertidor back to back se presenta en la Figura 2.

El sistema puede funcionar de dos maneras, en el primer modo se traspasa una cantidad especificada de potencia real y reactiva entre la red y la microrred. En el segundo modo la demanda de la microrred es tan alta que los generadores trabajan al máximo y la potencia que debe pasar a través del convertidor no puede ser especificada, en este caso se pierde la utilidad del convertidor. Por último se concluye tras varias pruebas con diferentes cargas que es un sistema eficaz a la hora de: proporcionar aislamiento de frecuencia, proporcionar aislamiento de voltaje y controlar el flujo de potencia.

En [6] se informa acerca de la composición de las microrredes destacando que tienen componentes de DC y componentes de AC, llegando a la conclusión de que esto puede generar diversos problemas. Como por ejemplo la necesidad de flujo de potencia de DC a AC, la estabilidad de las tensiones o la calidad y seguridad del suministro a las cargas. Para solucionar estos inconvenientes se presenta la estructura de una microrred híbrida gracias al convertidor back to back.

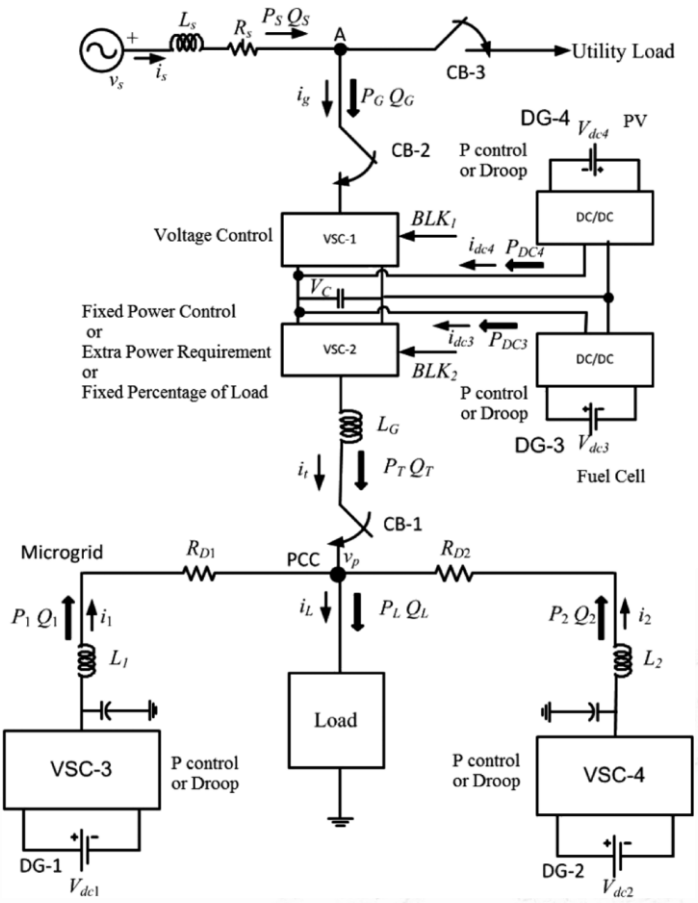

Figura 3. Estructura de la microrred de [6].

La estructura de la microrred propuesta tiene varias partes, véase la Figura 3. La parte alterna de la microrred contiene dos generadores y una carga, los generadores se conectan a la microrred mediante convertidores (VSC). Esta parte se conecta a la red de distribución a través del convertidor back to back. La parte que más destaca es el nudo de corriente continua donde se han conectado otros dos generadores mediante unos convertidores DC/DC, que ajustan la tensión al valor del nudo del convertidor back to back.

Se resalta que el convertidor back to back puede proporcionar aislamiento respecto de la red de distribución, la conexión eficaz de generación distribuida de corriente continua y

la posibilidad del paso de potencia de las zonas de corriente alterna a las zonas de corriente continua. Además se estudia cómo reacciona el sistema con varios tipos de controles y bajo diferentes circunstancias deduciéndose que la estructura propuesta es eficaz.

En [7] se dice que los convertidores pueden aislar a una microrred de las fluctuaciones de la red de distribución. Pero que la protección de la microrred, las cargas desequilibradas y los armónicos pueden generar problemas si la microrred es alimentada por un convertidor. Se presenta el caso de un pueblo de Finlandia que tenía problemas con el suministro, para solucionarlo se instaló un inversor back to back de 120 KVA que incluye filtros LCL. El artículo presenta las primeras medidas de este tipo de convertidor funcionando en una red real.

El inversor se divide en dos partes, la parte del convertidor en contacto con la red se encarga de las rectificaciones y la parte conectada a la microrred regula la frecuencia y el voltaje. Además, se puede monitorear remotamente mediante comunicación inalámbrica GPRS. El sistema de comunicación tiene una batería de reserva para los apagones.

Las variaciones de tensión de la red de distribución no afectaron a la tensión de la microrred ni tampoco los cambios bruscos en el consumo. La distorsión total armónica se mantuvo por debajo del 3%. También se comprobó el funcionamiento de los fusibles de protección mediante ensayos de cortocircuitos. Se observó que los condensadores del convertidor fueron suficientes para soportar pequeñas interrupciones en el suministro.

1.1.2 ALGORITMOS DE CONTROL

En la Tabla 2 se puede ver una comparación de los diferentes trabajos considerados en este estudio del estado del arte relacionados con el control o la simulación de microrredes.

Trabajo	*Fuentes de energía*	*Almacenamiento*	*Tipo de estudio*	*Tamaño*	*Algoritmo*
[8]	Solar fotovoltaica	Baterías	Simulación	35 KW	Precio energía
[9]	Solar fotovoltaica y unidad de cogeneración	Baterías 50 KWh	Control	>50 KW	Programación entera mixta CPLEX
[10]	Solar fotovoltaica y eólica	Baterías 35 KWh	Control	Caso 1 33,5 KW Caso 2 42.5 KW	Agentes
[11]	-	-	Control	-	Arquitectura distribuida
[12]	-	-	Control	-	Programación entera mixta
[13]	Solar fotovoltaica y unidad de cogeneración	Baterías 50 KWh	Control	> 50 KW	Árbol de los posibles estados CPLEX
[14]	Solar fotovoltaica, eólica y generador diesel	Baterías 10 KWh	Control	15 KW	1º Renovables 2º Baterías 3º Generador diesel
[15]	-	-	Control	11 KW	Prioridad de las cargas y nivel de comodidad
[16]	Solar fotovoltaica, eólica y generador diésel	Baterías 10.8 KWh	Simulación	50 KW	Cooperativo Simulación en lazo cerrado

Tabla 2. Comparativa relacionada con el control de la microrred.

A continuación se comentan detalladamente cada uno de los trabajos analizados.

En [8] se empieza con unas definiciones de ciertos conceptos. La parte principal consiste en la simulación de una red con cinco edificios inteligentes con: una carga eléctrica, generación de energía solar fotovoltaica y almacenamiento en baterías. El programa de respuesta a la demanda se puede basar en el precio de la energía en cada momento o en el precio de la energía y en un control directo de ciertas cargas por parte de la compañía eléctrica. Para simular la variabilidad de la respuesta de los cinco edificios, debido a las diferentes estrategias de los usuarios, se usa una distribución de probabilidad uniforme.

Así mismo, se crean cinco escenarios diferentes para validar el correcto funcionamiento de la simulación. Analizando los resultados de dicha simulación se deduce que si la compañía tiene el control de ciertas cargas el consumo es bastante menor que si el programa de respuesta a la demanda se basa solo en el precio de la energía en cada momento. Las conclusiones a las que se llega son que el modelo actuaba como se

esperaba y que realmente se producirá un gran ahorro de energía cuando se implante un sistema similar a gran escala en una ciudad.

En [9] se plantea el problema de establecer la estrategia óptima en el control de una microrred para minimizar el coste global de la electricidad y el gas consumidos para el correcto funcionamiento del edificio. Las fuentes de energía de la red serán unos paneles fotovoltaicos y un sistema de cogeneración. Además tiene conexión con la red y unas baterías de almacenamiento de energía eléctrica.

Se asume que el edificio tiene contadores inteligentes (Smart meters), es decir, contadores que miden el consumo o generación cada hora y envían esa información a un sistema central. Al plantear el problema se obtiene un problema no lineal debido a las características de la función y las restricciones. Se crean algunas variables enteras para linealizar un poco el problema y se aplica un método de programación entera mixta para resolverlo. El problema se resuelve mediante CPLEX y se tiene en cuenta el precio de la electricidad a cada hora del día.

Se hicieron dos escenarios uno con baterías y otro sin ellas, como se esperaba el ahorro es mayor en el escenario que incluye el sistema de almacenamiento de energía. El coste total se ve reducido en un 6.5% respecto al sistema sin almacenamiento energético. Tras varias pruebas numéricas se concluye que el problema de optimización entero mixto planteado, junto con el optimizador CPLEX, son una solución efectiva al problema planteado.

En [10] se presenta uno de los problemas que se generan en las microrredes, debido al compromiso entre la reducción del consumo energético y la necesidad de mantener el nivel de comodidad de los usuarios. En este análisis se concentra en tres aspectos: la temperatura, el nivel de iluminación y la calidad del aire.

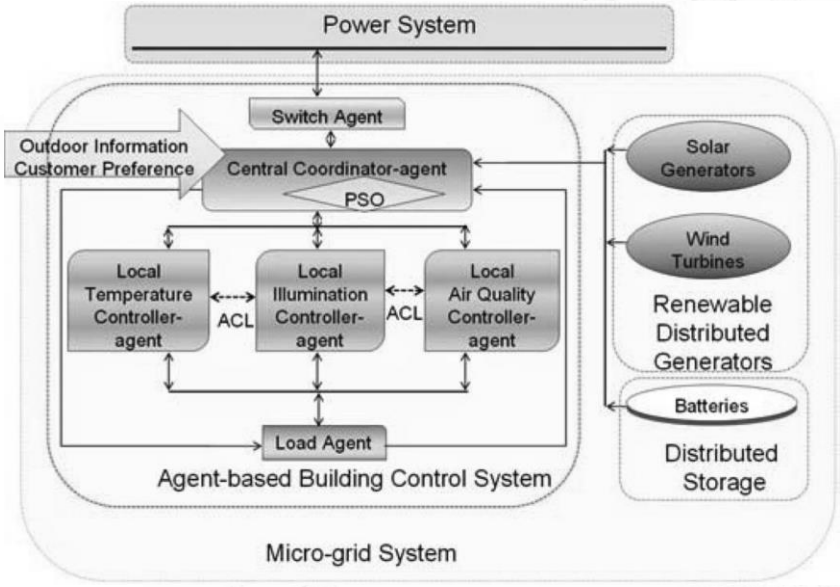

Figura 4. Arquitectura del control y sistema de gestión de [10].

Como solución al problema se presenta el diseño de un control con cuatro tipos de agentes que está jerarquizado, véase la figura 5, basando en métodos heurísticos de optimización. Hay un agente (switch agent) encargado de determinar si la microrred se conecta a la red de distribución o no. Un segundo agente (Central coordinator agent) se encarga de decidir la energía que se aporta a cada uno de los agentes locales y al agente de la carga. Los agentes locales (local agent) son tres: uno para la temperatura, otro para la iluminación y otro para la calidad del aire.

Estos agentes controlan las cargas que influyen directamente en el aspecto que regulan para maximizar la comodidad con la energía que les ha aportado el agente que coordina la microrred. El agente de la carga controla el resto de las cargas. Como conclusión se aprecia que el sistema funciona correctamente cumpliendo los objetivos al coordinar los distintos agentes.

En [11] se menciona que las redes de sensores inalámbricas están volviéndose más importantes, sobre todo a la hora de las comunicaciones en tiempo real usadas en las redes inteligentes. El problema reside en que los sistemas de control no son implementados eficazmente en las casas. Esto se debe a las limitaciones de su arquitectura y a las baterías de los sensores. Se propone un sistema inteligente que se adapta automáticamente ahorrando energía mediante el control de cargas.

Esto se consigue pasando del sistema clásico con una arquitectura centralizada, hay un servidor que recibe la información y hace todo el procesamiento, a un sistema con una arquitectura distribuida en el que hay varios puntos de procesamiento (SIG). Cada centro de procesamiento de información se adapta al contexto en el que se encuentra comunicándose con los sensores próximos mediante una red inalámbrica de sensores (SIS). Posteriormente, se comprobaron los resultados en un banco de pruebas viéndose que el consumo se redujo entre un 16% y un 24%.

En [12] se presenta una arquitectura diseñada para el control de las cargas en edificios inteligentes, que permite al consumidor controlar dichas cargas de manera autónoma. Esta arquitectura presenta una solución para controlar las cargas internamente sin que esté implicada la empresa distribuidora de energía. La arquitectura, véase la Figura 5, está compuesta de tres niveles: el control de admisión, el equilibrio de la carga y la gestión de la demanda.

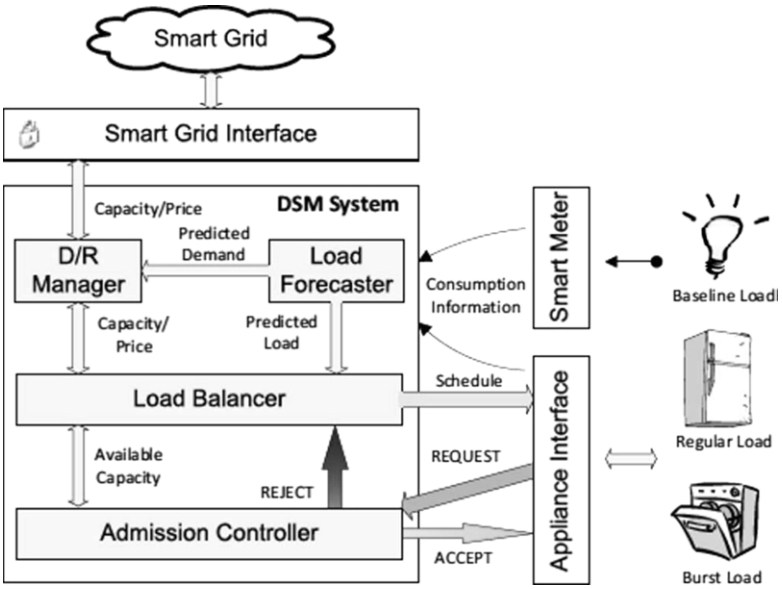

Figura 5. Arquitectura del control y sistema de gestión de [12].

Se han desarrollado dos de los tres niveles: el control de admisión y el equilibrio de la carga. El controlador de admisión genera una planificación en tiempo de ejecución, que controla el funcionamiento de las aplicaciones, satisfaciendo un cierto nivel de comodidad respetando el consumo máximo. Por otro lado, el equilibrador de carga resuelve un problema de programación lineal entera mixta minimizando la función de coste teniendo en cuenta los tiempos límite de las tareas y el consumo máximo permitido por el gestor de la demanda. Como el gestor de la demanda no está desarrollado, la arquitectura es válida para muchos casos incluyendo incluso microrredes y controles jerárquicos.

En [13] se informa acerca de un estudio que revela que se podría ahorrar entre el 20% y el 30% de la energía consumida en un edificio. En este trabajo se desarrolla una solución para el problema de la planificación de las fuentes de energía usadas en cada momento dentro de una microrred, con varias fuentes de energía y baterías de almacenamiento.

El objetivo es minimizar el coste de la energía necesaria para el funcionamiento del edificio a lo largo del tiempo dando servicio a los usuarios. Se debe tener en cuenta que existen incertidumbres acerca de la generación de las fuentes renovables y del consumo de los usuarios. Para solucionar este problema se usa el método del árbol de los posibles estados y posteriormente se plantea un problema de programación lineal entera mixta que se soluciona mediante CPLEX. Además, se efectúan varias pruebas numéricas y se comprueba que realmente funciona el sistema propuesto.

Por último, se concluye que es importante usar toda la energía solar, optimizar la gestión de la batería y que el enfriamiento previo durante los periodos de precio bajo de la energía implica un ahorro importante.

En [14] se introduce el concepto de nanored (nanogrid) que consiste en una red de corriente continua aislada de la red de distribución. Posteriormente se indica que el objetivo del trabajo es el diseño de un sistema de control de este tipo de redes. El sistema de control se basa en medir la tensión en cada uno de los nodos de la red y en función de estos valores determina el estado en que se encuentra la nanored. La estrategia del control se basa en usar en primer lugar las fuentes renovables de energía y si sobra generación recargar las baterías. Si la tensión baja, las fuentes renovables no aportan suficiente energía, se usan las baterías. Únicamente en el caso de que esto no sea suficiente se activan las fuentes de energía no renovables. Por último se realiza una simulación que comprueba la viabilidad del sistema de control.

En [15] se comenta que los programas de respuesta a la demanda, que controlan el consumo del usuario para evitar los picos del consumo, se han implementado en consumidores industriales pero no en los hogares. Se presenta un sistema de control (Home Energy Management) para gestionar el consumo de una casa. El objetivo del algoritmo es mantener el consumo por debajo de un límite durante ciertas horas, todo ello fijado por el usuario. El algoritmo de control se basa en la prioridad de cada carga y en intentar mantener el nivel de comodidad del usuario, véase la Tabla 3.

Appliance	Load Priority	Homeowner Preference
Water Heater	1	Water Temperature: 110-120°*F*
Space Cooling Unit	2	Room Temperature: 76°*F* (±2°*F*)
Clothes Dryer	3	Finish the job by midnight Maximum OFF time: 30 mins Minimum ON time: 30 mins
EV (Level 2 charging w/ 240V single phase)	4	Fully charged by 8 AM Minimum charge time: 30 mins

Tabla 3. Prioridades de las cargas y preferencias en [15].

En primer lugar se recopila información acerca del estado de las cargas, del consumo máximo permitido y del tiempo que dura la restricción. Seguidamente se comprueba si se alcanzan los niveles de comodidad. En tercer lugar si no se alcanza algún nivel se comprueba si se puede encender la carga correspondiente. Seguidamente se comprueba si se supera el límite del consumo. Si se supera este límite se van apagando las cargas de menor a mayor prioridad hasta estar por debajo del límite. Como conclusiones por un lado se comprueba mediante simulación que el sistema es efectivo, y por otro lado, se advierte que es posible que se genere un pico en el consumo una vez se supere el tiempo de control fijado por el usuario.

En [16] se menciona el desarrollo de varios tipos de controles para las microrredes que optimizan el coste y diversos tipos de simulación de controles. Pero es difícil comparar adecuadamente los tipos de controles ya que, en general, se implementan en redes distintas. En este artículo se desarrolla un hardware para simular en lazo cerrado diferentes tipos de controles de microrredes. El sistema se compone de: un simulador digital en tiempo real de la microrred, el sistema de control que se esté poniendo a prueba y un emulador de las comunicaciones con los distintos componentes de la microrred.

La microrred simulada en tiempo real tiene como fuentes de energía unos paneles fotovoltaicos, una turbina eólica y un generador diésel, véase la Figura 6. Además posee unas baterías de 10.8 KWh, dos cargas y conexión a la red de distribución.

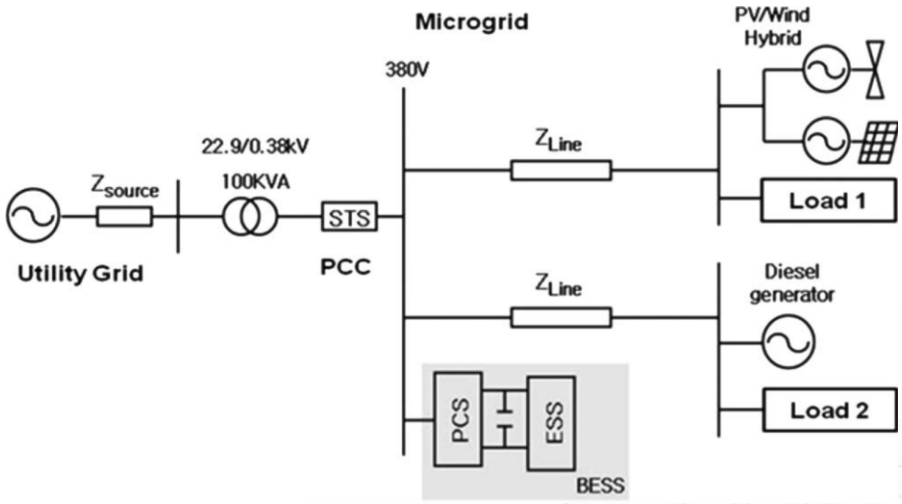

Figura 6. Esquema de la microrred de [16].

Posteriormente se comprueba el funcionamiento de un control cooperativo en el que el sistema de almacenamiento de las baterías (BESS) regula el flujo de potencia en el punto de acoplamiento común (Figura 6) y luego el sistema de control calcula la producción necesaria del generador diésel teniendo en cuenta la potencia de las baterías y de PV y eólica. Además en el modo isla de la microrred, el control se encarga de regular la tensión y la frecuencia de la tensión. Se concluye que el hardware de simulación en lazo cerrado ha sido validado con éxito al comprobar su funcionamiento aplicándole un control cooperativo de microrredes.

1.1.3 SISTEMAS DE COMUNICACIÓN

En la Tabla 4 se puede ver una comparación de los diferentes trabajos considerados en este estudio del estado del arte relacionados con los sistemas de comunicación de las microrredes.

Trabajo	*Fuentes de energía*	*Almacenamiento*	*Método de comunicación*	*Tamaño*
[17]	-	-	Red IP fibra óptica o WiMax	-
[18]	Dos generadores	Baterías	La tensión y la frecuencia de la microrred	18 KW
[19]	Solar fotovoltaica, eólica y pila de combustible.	Baterías	Red IP Wi-Fi modo infraestructura	-
[20]	Solar fotovoltaica, eólica y pila de combustible.	Baterías	eCAN para las ordenes Modbus para los datos	-
[21]	-	-	Bluetooth, Zig Bee y LAN	-

Tabla 4. Comparativa de los sistemas de comunicación.

A continuación se comentan detalladamente cada uno de los trabajos analizados.

En [17] se indica que las redes inteligentes del futuro se caracterizarán no solo por un flujo bidireccional de potencia sino por dicho flujo y por una comunicación bidireccional de todos los puntos de la red. Además se destaca que el elemento más importante de las redes inteligentes será el medidor inteligente (Smart meter). Entre datos directos y datos calculados se estima que será necesario enviar 5 Mbits/s aproximadamente desde los puntos de control. Se revela que usar una red de comunicaciones con base de IP sería lo más conveniente. Además, la extrema sensibilidad del sistema al paso del tiempo exige una latencia máxima de 100 milisegundos. Teniendo estos dos factores en cuenta el sistema más apropiado es la fibra óptica ya que tiene una latencia muy baja del orden de 5 microsegundos por kilómetro. Cuando no sea posible usar fibra óptica se recomienda el uso de WiMAX.

WiMAX es una tecnología inalámbrica con una latencia muy baja, normalmente inferior a los 10 milisegundos. El núcleo del sistema de comunicación, responsable de unir los elementos de la red de WiMAX, deberá mantener su latencia por debajo de los 50 milisegundos para obtener cierto margen. Por último se concluye que las redes inteligentes deberán tener un sistema de comunicación para satisfacer las necesidades futuras y que solo incluyendo la generación distribuida de energía se podrá generar realmente una red inteligente de energía eléctrica.

En [18] se señala que las microrredes deben poder funcionar en modo isla y conectadas a la red de distribución y que son posibles gracias a diversos factores, entre ellos la generación distribuida. Uno de los problemas es mantener el voltaje y la frecuencia en niveles aceptables en el modo isla y para ello es necesario establecer una comunicación y tener un sistema de almacenamiento. En ocasiones se usan métodos de comunicación que emplean conexiones separadas. En cambio en este artículo se presenta un sistema de comunicación que usa el voltaje y la frecuencia de la microrred como parámetros de la comunicación. La microrred consta de dos generadores, una carga y un sistema de almacenamiento.

Se utiliza un método de control de caída para regular la microrred siendo capaz de funcionar en dos modos: el modo isla y el modo conectado a la red. En modo isla la frecuencia y el voltaje están fijados por la acción de la batería. Cuando la frecuencia baja la batería aporta potencia activa subiendo la frecuencia y si baja el voltaje aporta potencia reactiva. Por otro lado si la frecuencia y la tensión están bien y la batería tiene poca carga (<20%), esta pasa del modo descarga al modo carga. Se toman dos corrientes de referencia, se calcula el error y utilizando un control PI se generan las señales de control del PWM del inversor de la batería, véase la Figura 7.

Por último, tras diferentes simulaciones se comprueba que el sistema mantiene la frecuencia y la tensión en márgenes aceptables.

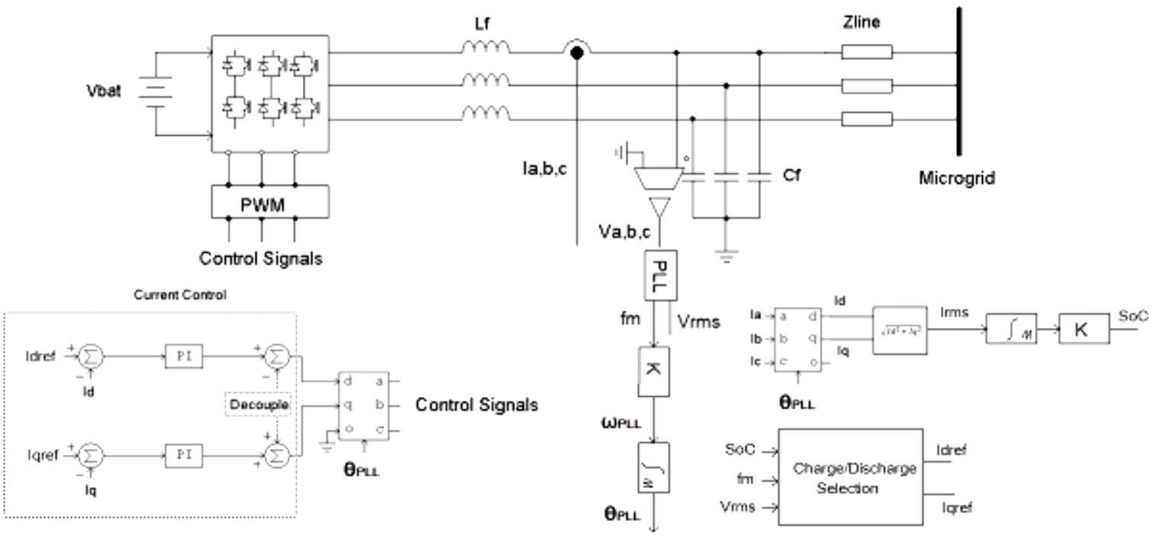

Figura 7. Esquema de la microrred de [18].

En [19] se describe el concepto de microrred y se explica que para hacer funcionar la microrred se necesita un sistema de gestión que se base en una infraestructura de comunicaciones. Este artículo se centra en el desarrollo de una red de comunicaciones Wi-Fi para el control de una microrred que puede instalarse en un laboratorio o en una casa. Se desarrolla un sistema de detección y control mediante MENS. Los MENS son programas de ordenador que tienen funciones de control y de adquisición de datos que se conectan a las fuentes de energía y a las cargas. Las redes Wi-Fi pueden funcionar en modo infraestructura o en modo ad-hoc. En el modo infraestructura los dispositivos inalámbricos se conectan a la red cableada a través de un punto de acceso. En el modo ad-hoc los dispositivos inalámbricos se conectan entre si directamente. El modo infraestructura tiene las ventajas de una gestión centralizada de la seguridad y mayor cobertura. En este caso se implementó una red en modo infraestructura y se usó un router inalámbrico como punto de acceso, véase la Figura 8.

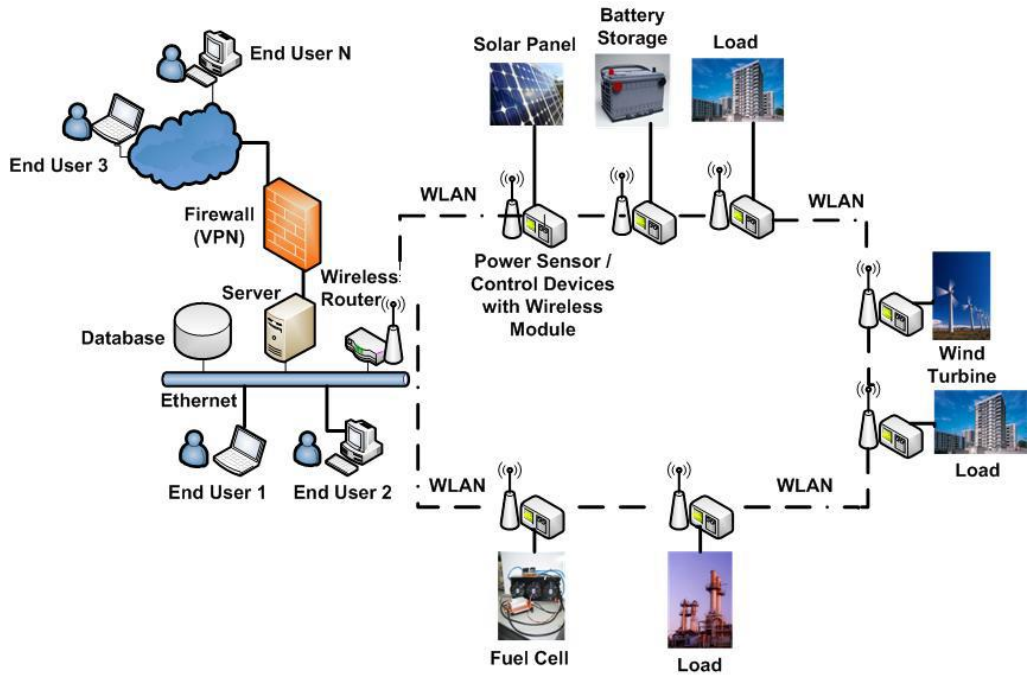

Figura 8. Esquema del sistema de comunicación de la microrred de [19].

Se concluye que aunque la red Wi-Fi es más lenta es una solución apropiada para microrredes pequeñas debido a una instalación fácil, un coste bajo de mantenimiento y sobre todo debido a su flexibilidad a la hora de incluir más dispositivos.

En [20] se recuerda el concepto de microrred y los dos modos de funcionamiento de esta. Además se indica la necesidad de un buen sistema de comunicación para obtener eficiencia en la microrred. La microrred propuesta está diseñada para funcionar con un control maestro-esclavo. El inversor principal conecta la línea de DC con la línea de AC. En la línea de AC se conectan los generadores, las cargas y la red de distribución. En la línea de DC se conectan las baterías, véase la Figura 9.

Cuando la microrred está conectada a la red el inversor principal funciona como cualquier otro intercambiando potencia, pero cuando la microrred funciona en modo isla es el encargado de controlar la tensión y la frecuencia de la microrred. El objetivo en modo isla es que la microrred funcione el mayor tiempo posible y, en cambio, el objetivo cuando está conectada a la red es mejorar la eficiencia económica. Las órdenes que se deben transmitir por parte del sistema para controlar la microrred son 4 bytes, mientras que los datos que se deben recibir son 20 bytes.

Se pensó en utilizar un módulo de comunicaciones eCAN pero no tiene suficiente ancho de banda para enviar los datos. Otro posible método de comunicación es el protocolo Modbus que posee suficiente ancho de banda, el problema reside en que es un sistema más lento. Por ello se decidió desarrollar una plataforma de comunicación múltiple, usando el eCAN para enviar las órdenes y el Modbus para recibir los datos del estado de

la microrred. Por último se comprobó que la plataforma de comunicaciones funcionaba correctamente.

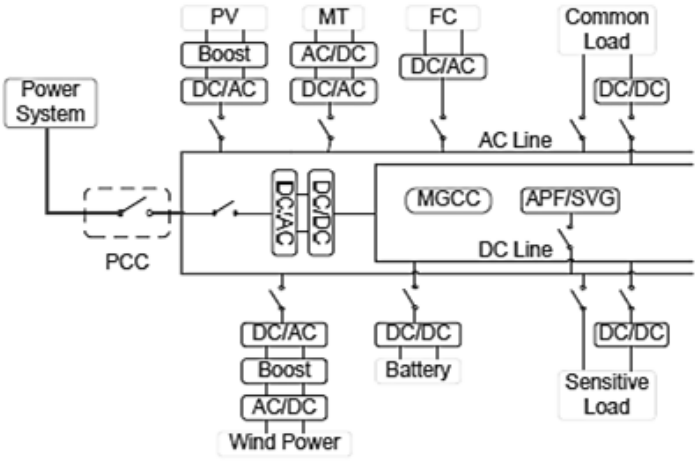

Figura 9. Esquema de la microrred de [20].

En [21] se elabora un desarrollo teórico de las partes de una red doméstica. Se señala que las redes domésticas se componen de redes de comunicación, diversidad de dispositivos, programas de control y un punto de acceso a internet. Las redes domésticas poseen una jerarquía de cuatro niveles: la red de área corporal, la red personal, la red de área local y la red de área amplia (WAN). Véase la Figura 10. La red de área corporal es una red inalámbrica de corto alcance que conecta pequeños dispositivos del cuidado de la salud. La red personal es una red inalámbrica que tiene algo más de alcance que conecta distintos dispositivos en un radio de unos 10 metros (bluetooth, ZigBee) La red de área local es la red doméstica (LAN) y la red de área amplia es lo que comúnmente se denomina internet.

El punto de acceso conecta la red doméstica con internet y es el encargado de conectar las redes de nivel inferior entre sí, teniendo que realizar conversiones de un protocolo de comunicación a otro. Para realizar esta conversión, el punto de acceso necesita adaptadores de red. Se concluye que un hogar inteligente (Smart home) consiste en varios componentes tecnológicos y funciones de gestión. La estandarización de estas tecnologías contribuye a la realización de una red doméstica robusta. Este artículo muestra la arquitectura de un punto de acceso (home Gateway) como un posible método de implementación. El sistema de gestión es un componente crucial para conseguir una red doméstica ecológica y el punto de acceso propuesto será de ayuda para conseguirlo.

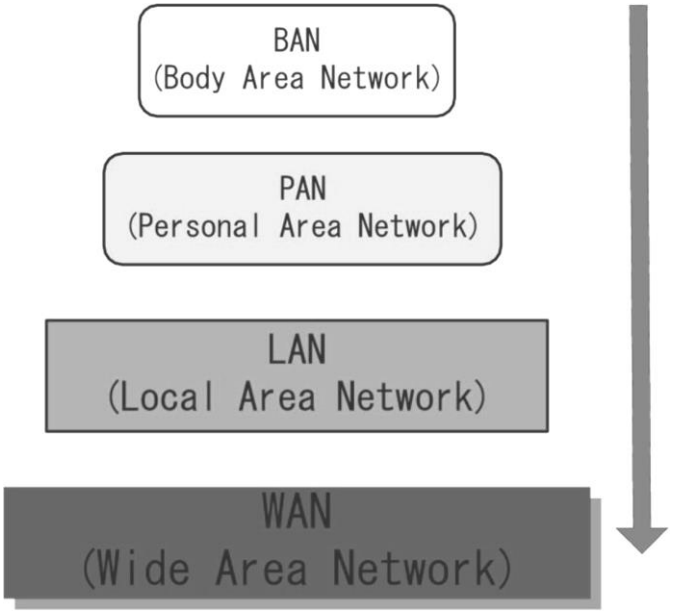

Figura 10. Niveles de una red domestica [21].

1.2 MOTIVACIÓN DEL PROYECTO

En esta sección se explica la motivación del proyecto y porque se ha desarrollado el proyecto de la manera escogida. Tras el estudio de las tecnologías existentes se observa que uno de los componentes principales son los convertidores DC/AC y los convertidores AC/DC. Se eligió trabajar con convertidor DC/AC por varios motivos: los convertidores DC/AC son más complejos que los convertidores AC/DC y además son los encargados de fijar las características de la tensión de la microrred, cuando esta se encuentra aislada de la red de distribución.

Cabe destacar que la mayoría de los inversores actuales no son reprogramables y que si lo son, es a un alto precio. Asimismo los inversores actuales, en general, tienen alguno de estos defectos: no se sincronizan con la red, no optimizan las condiciones de funcionamiento o necesitan una tensión base.

Por todo ello, la motivación principal es superar todos los defectos mencionados anteriormente intentando reducir el coste de fabricación.

1.3 OBJETIVOS

El objetivo principal del proyecto es consecuencia directa de la motivación del proyecto. Este objetivo es desarrollar un inversor (convertidor DC/AC) que cumpla los siguientes requisitos:

- Tener una potencia de salida de 200W.
- Tener la capacidad de ser reprogramable.
- Incluir elementos de optimización de las condiciones de funcionamiento.
- Estar descompuesto en varios módulos.
- Poder funcionar conectado a una red de distribución eléctrica y, al mismo tiempo, poder funcionar de manera aislada.
- Tener un coste inferior a 2000 €.

1.4 METODOLOGÍA

En primer lugar se tuvo que decidir que diseño se iba a utilizar como base o si se realizaría un diseño desde cero. Se decidió realizar el proyecto a partir de un diseño ya existente para poder abarcar el proyecto en un año. Tras un análisis de las características de varios inversores, se decidió desarrollar el inversor a partir del inversor AN1444 de Microchip. Los principales motivos fueron: tiene una potencia de salida muy similar a la de los objetivos, optimiza la tensión de funcionamiento maximizando la potencia entregada y el diseño se puede descargar de manera gratuita de una página de Microchip.

Posteriormente se estudiaron las tareas necesarias para la realización del proyecto y se elaboró un cronograma para cumplir los objetivos a tiempo, véase la Figura 11. Las tareas principales son: realizar el estudio del arte, determinar los objetivos, buscar los componentes, estudiar el inversor original, realizar los diseños en Eagle de cada uno de los módulos, programar los cambios necesarios para el modo isla, realizar pruebas para comprobar el funcionamiento y redactar la memoria.

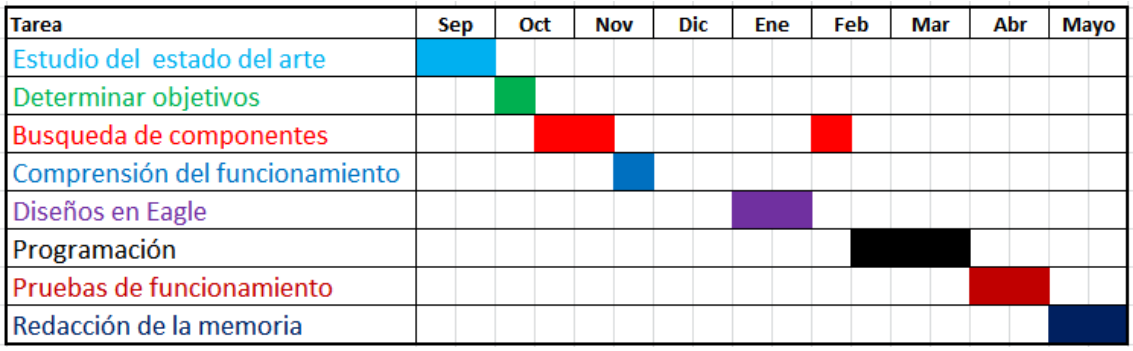

Figura 11. Cronograma del proyecto.

A continuación se explica brevemente lo que abarca cada una de las tareas.

- Estudio del estado del arte: esta tarea consiste en analizar los trabajos actuales relacionados con el proyecto a realizar. Se identifican los avances que se han ido realizando y se busca una manera de mejorar los sistemas actuales.

- Determinar los objetivos: es decidir y en hacer explícitos los objetivos que se desean conseguir.

- Búsqueda de componentes: consiste en buscar todos los elementos necesarios para construir el inversor objetivo, teniendo en cuenta los precios de cada proveedor.

- Comprensión del funcionamiento: es el análisis del hardware y software original. Así se podrá modificar el diseño original para conseguir las funciones añadidas deseadas.

- Diseños en Eagle: para reducir el coste del inversor se decidió pasar de las 4 capas del diseño original a un diseño en 2 capas.

- Programación: radica en desarrollar el software necesario para el funcionamiento del inversor en modo aislado.

- Pruebas de funcionamiento: consiste en probar el software desarrollado, verificando su correcto funcionamiento.

- Redacción de la memoria: es la elaboración del documento que recoge el trabajo realizado durante la ejecución del proyecto fin de carrera.

1.5 RECURSOS Y HERRAMIENTAS EMPLEADAS

En esta sección se indican las herramientas y los programas de ordenador que se utilizaron en la elaboración del proyecto.

Las herramientas que se emplearon fueron: un ordenador de sobremesa, un osciloscopio de 4 canales, una fuente de alimentación de 30V, sondas de osciloscopio, cables de conexión, una regleta de protección y herramientas varias (destornillador, alicates, tijeras, etc.)

Los programas a los que se recurrió fueron: Microsoft Word, Microsoft Excel, Microsoft Power Point, Eagle y MPLAB. Destacando que las versiones que se usaron de Eagle y MPLAB son software libre lo que ayudó a reducir el coste de la ejecución del proyecto.

Capítulo 2 ARQUITECTURA

2.1 INTRODUCCIÓN

Este capítulo tiene como propósito explicar el funcionamiento del inversor de manera superficial antes de realizar un estudio más profundo en los capítulos siguientes. La descripción del funcionamiento se dividirá en varias secciones: hardware, software general, algoritmos de control y algoritmo de rastreo del punto de máxima potencia (MPPT Maximun Power Point Tracker).

2.2 HARDWARE

En esta sección se indican los módulos presentes en el hardware del inversor AN1444, así como la relación entre ellos. Posteriormente se hará una descripción más detallada en el capítulo 3.

Los módulos principales que presenta el hardware son: los condensadores de desacoplamiento, el convertidor DC/DC, el inversor Full Bridge, el filtro EMI y el sistema de control del microprocesador. Para ver la relación entre ellos véase la Figura 12, en ella se representa un diagrama de bloques del microinversor AN1444.

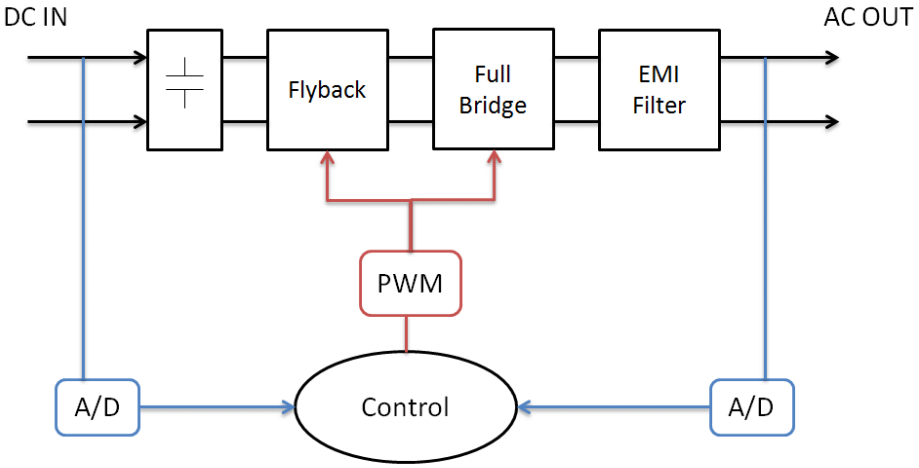

Figura 12. Diagrama de bloques del hardware del microconversor AN1444.

El primer módulo presente son los condensadores de desacoplamiento, su función es la de mantener la tensión constante ante variaciones de tensión instantáneas en la entrada. A continuación se encuentra el convertidor DC/DC, en realidad son dos convertidores Flyback que funcionan en paralelo. Estos además de aumentar el nivel de tensión proporcionan aislamiento galvánico entre la entrada y la salida.

Posteriormente se halla el inversor Full Bridge, este módulo se encarga de generar una onda alterna a partir de una tensión continua. Este módulo se controla mediante un PWM desde el microprocesador. Justo antes de la salida se encuentra el filtro EMI (Electromagnetic Interference filter) este módulo es el encargado de atenuar los armónicos superiores. Gracias a esta atenuación la onda alterna de salida es casi senoidal, además posee dos fusibles de protección.

Por último se puede observar el módulo de control, que incluye: el microprocesador, los circuitos de acondicionamiento de medidas, el conversor A/D y el PWM. Las funciones que desarrolla este módulo son el control del inversor y la protección del mismo, mediante varias verificaciones.

2.3 SOFTWARE

En esta sección se da una idea general del código del microprocesador encargado de controlar el inversor, el código se analizará más profundamente en el capítulo 4.

Toda la información le llega al microprocesador mediante la interrupción del conversor A/D, exceptuando la señal del comparador. Esta señal indica si hay una sobre corriente en el convertidor Flyback, lo que implica un error grave.

Una vez recibidas las medidas se realizan varias verificaciones de control y se ejecuta la máquina de estados programada, véase la Figura 41. Durante las verificaciones se comprueba el valor de: la tensión de entrada, la corriente de entrada, la tensión de salida, la corriente a la salida, la corriente del Flyback, etc. Dependiendo del estado en el que se encuentre el programa se generará la respuesta adecuada. Todas las salidas son señales de control digitales y señales PWM.

Respecto a la máquina de estados hay tres estados: el estado de error, el estado de arranque y el estado de funcionamiento. El programa está en todo momento en un estado.

En el estado de error se comprueba si hay errores. Si los errores desparecen espera durante 1 segundo y pasado este tiempo se pasa al estado de arranque.

En el estado de arranque se verifica que no se da ningún error y, si es así durante 60 cruces por cero de la tensión de referencia, se salta al estado de funcionamiento. Si se produce algún error se vuelve al estado anterior.

En el estado de funcionamiento se activa el inversor Full Bridge, se comprueba que no haya ningún error y se actúa sobre las señales de mando para que la onda generada esté sincronizada con la red. Si se produce algún error se vuelve al primer estado.

2.4 ALGORITMOS DE CONTROL

En esta sección se comentan los lazos de realimentación utilizados para controlar el microinversor AN1444. En este diseño coexisten tres lazos de realimentación, véase la Figura 43.

Existe un control para repartir la carga entre los dos convertidores Flyback que están en paralelo. Se trata de un control proporcional integral que mide las corrientes de salida de cada Flyback y calcula el incremento o decremento del factor de servicio de cada Flyback. Al tener dos convertidores en paralelo con la mitad de corriente cada uno se reducen las pérdidas, mejorando el rendimiento.

El factor de servicio de los convertidores Flyback se calcula mediante otro lazo de realimentación con un control proporcional integral. Como datos de entrada usa la tensión de entrada, la tensión pico de salida, la corriente de salida y los incrementos calculados

por el lazo mencionado anteriormente. Calcula el factor de servicio de ambos convertidores DC/DC para obtener el nivel de tensión deseado a la salida. Esto permite tener una tensión de 230 Va la salida aunque la tensión de entrada varíe entre 25 y 45 V.

Finalmente existe un lazo de realimentación que sincroniza la tensión de salida con la tensión de la red. Este control proporcional mide el momento de tensión pico de la red y actúa sobre el PWM que controla el Full Bridge.

2.5 ALGORITMO MPPT

En esta sección se describe el funcionamiento del algoritmo de rastreo del punto de máxima potencia (*MPPT Maximun Power Point Tracker*) El objetivo de este algoritmo es optimizar la potencia entregada buscando la tensión de entrada que la maximiza.

Este algoritmo es especialmente útil si conectamos un panel solar a la entrada del inversor. Los paneles solares tienen una tensión a la que la potencia entregada es máxima. Dicha tensión depende de la temperatura y la luminosidad, véase la Figura 13.

Este algoritmo usa el método de perturbación y observación (*P&0 Perturb and Observe Method*) Se genera una pequeña perturbación, si esta perturbación aumenta la potencia la siguiente perturbación tendrá el mismo signo. Por el contrario, si se disminuye la potencia entregada, la siguiente perturbación tendrá signo contrario. Este algoritmo se ejecuta una vez cada tres periodos de la onda alterna, por lo que el punto de trabajo va variando constantemente.

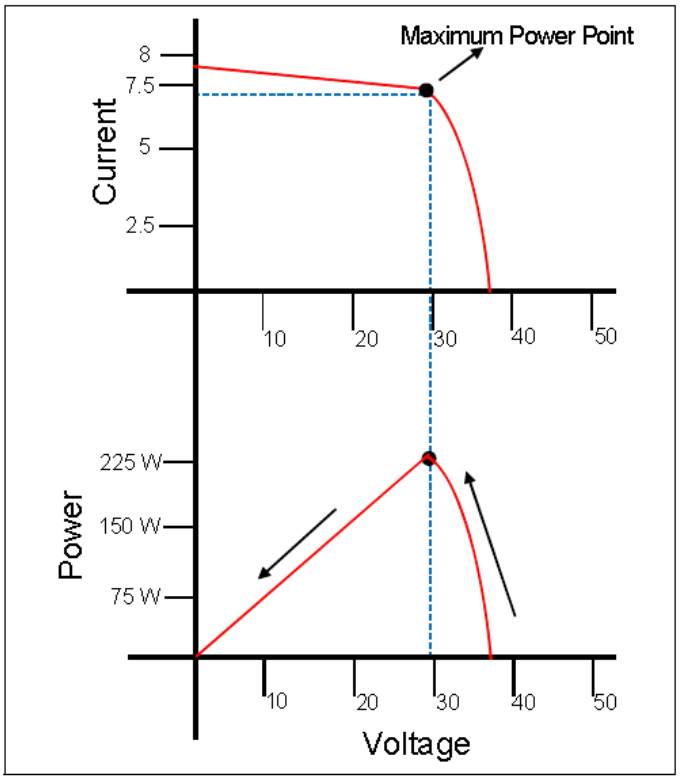

Figura 13.Funcionamiento del algoritmo MPPT.

Capítulo 3 ANÁLISIS DEL HARDWARE

3.1 INTRODUCCIÓN

En este capítulo se indican los diseños de los circuitos analógicos que sirven para medir, actuar o controlar el inversor. Así mismo, se muestran los diseños del circuito impreso de cada uno de ellos.

3.2 SENSOR DE CORRIENTE ALTERNA

En esta sección se describe el circuito de medida de corriente alterna de la salida, véase el esquema del circuito en la Figura 14. Este circuito de acondicionamiento toma la señal del componente U14 y la modifica para que la lea correctamente el microprocesador. El componente U14 es un sensor de corriente con un offset de 2.5 V y una sensibilidad de 180 mV/A. La señal de salida de este circuito se caracteriza por tener una tensión de 1.65 V para corriente nula y 3.3 V de tensión máxima. La ecuación que describe este circuito se puede observar en la Ecuación 1.

$$V_{AC_CURRENT} = 2 \cdot \left[5 \cdot \left(\frac{R37}{R37 + R35}\right) \cdot \left(\frac{R28}{R28 + R34}\right) - I_{out}\right] \cdot \left(1 + \frac{R21}{R20}\right)$$

Ecuación 1. Ganancia del circuito del sensor de corriente alterna.

Para realizar estas operaciones utiliza dos amplificadores operacionales que en este caso están formados por un mismo componente U5. El amplificador operacional denominado U5A toma un valor de tensión constante y le resta un valor de tensión proporcional a la corriente de entrada. Por otro lado el operacional U5B amplifica la señal mediante una etapa de ganancia. Se añadió un condensador para estabilizar la tensión. Por último el diodo doble D7 se utiliza para mantener la tensión entre 0 y 3.3 V, este elemento se añadió para proteger al microprocesador ante una posible sobretensión y ante tensiones negativas.

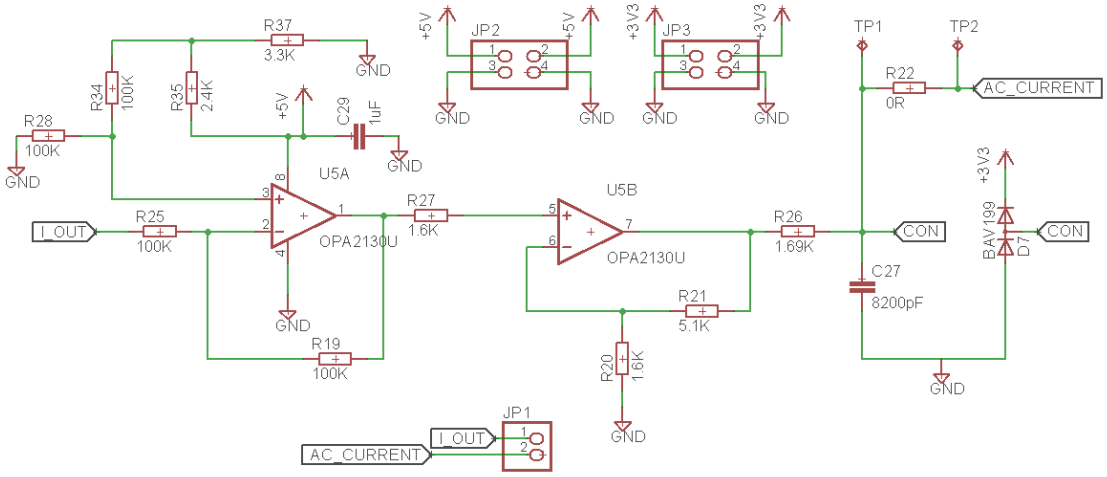

Figura 14. Esquema del sensor de corriente alterna.

A continuación se muestra el circuito impreso diseñado, que está representado en la Figura 15.

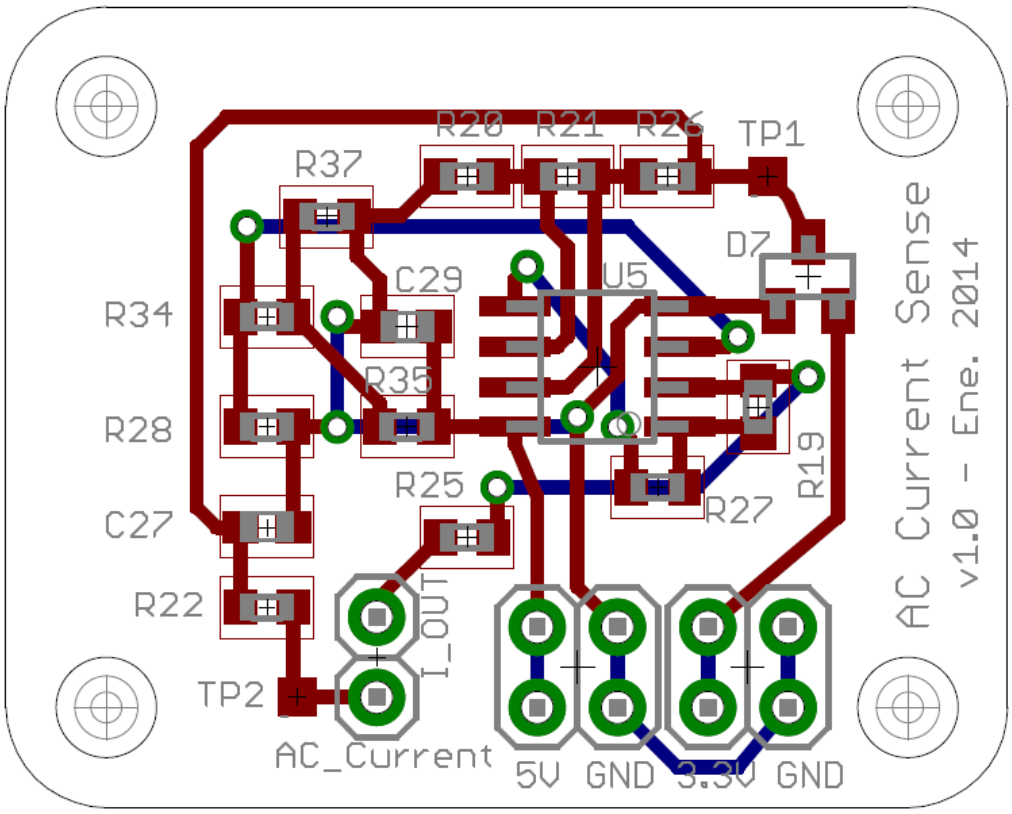

Figura 15. Circuito impreso del sensor de corriente alterna.

3.3 SENSOR DE TENSIÓN DE ENTRADA

En esta sección se detalla el circuito encargado de medir la tensión de entrada, en este circuito también se mide la temperatura del inversor y se genera la tensión de referencia de 2.5 V. El circuito de acondicionamiento de la tensión de entrada se realizó mediante un divisor de tensión, véase la Figura 16. Se añadió un condensador para estabilizar la tensión y un diodo doble de protección, que impide que la tensión sea superior a 3.3 V o inferior a 0 V. La referencia de 2.5 V se realizó mediante un diodo Zener alimentado a 5V y la tensión proporcional a la temperatura se obtiene directamente del sensor alimentado a 3.3V.

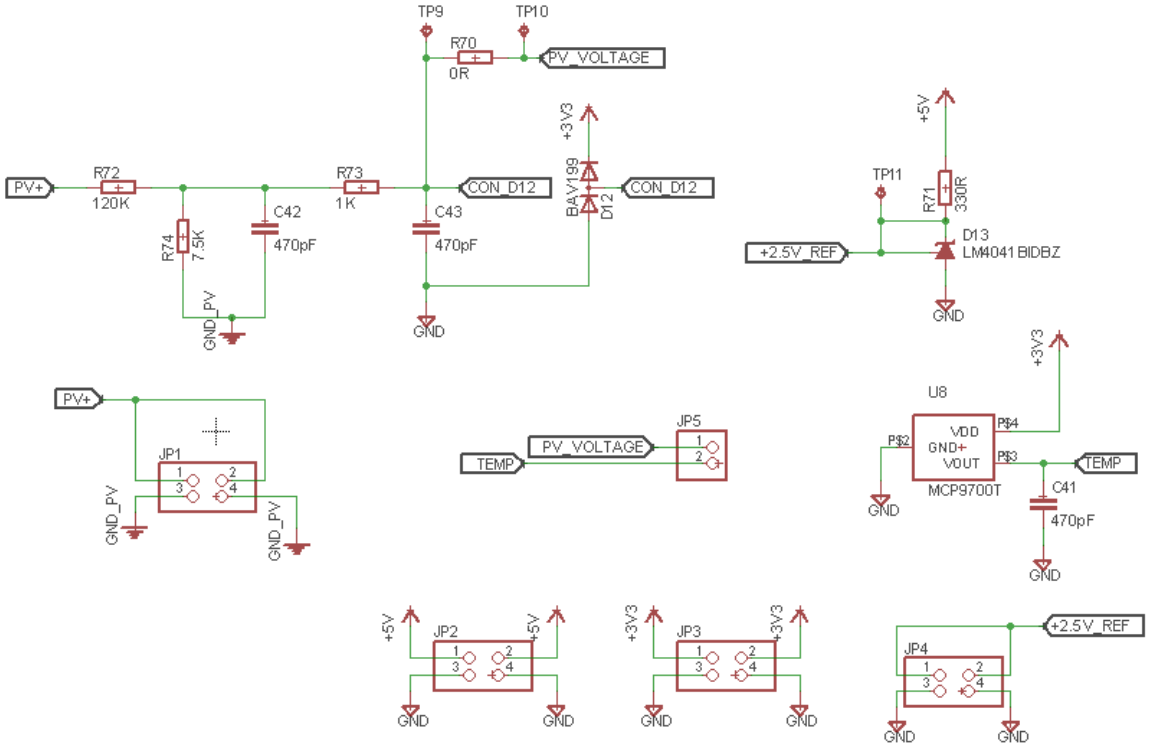

Figura 16. Esquema del sensor de tensión de entrada.

A continuación se muestra el circuito impreso diseñado, que está representado en la Figura 17.

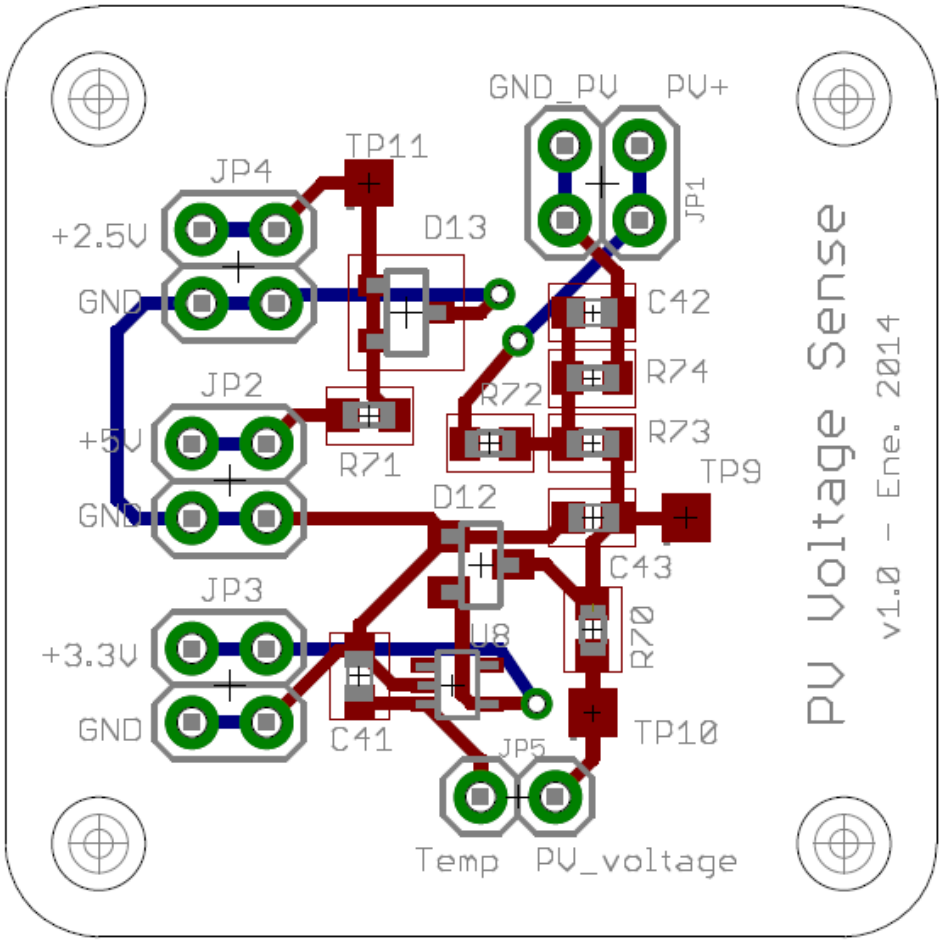

Figura 17. Circuito impreso del sensor de tensión de entrada.

3.4 SENSOR DE TENSIÓN ALTERNA

En esta sección se describe el circuito del sensor de tensión alterna, cabe destacar que también se genera una señal para indicar los cruces de la tensión con cero. Para medir la tensión alterna se realiza una reducción con un transformador de 230 V a 5 V, se le suma la referencia de 2.5V y se realiza otra reducción mediante el amplificador U6. La siguiente ecuación describe las operaciones realizadas por el circuito.

$$V_{AC_VOLTAGE} = \left[\left(\frac{V_{grid}}{N_{TR1}} \cdot \frac{R36}{R31}\right) + 2.5\right] \cdot \frac{R32}{R30 + R32}$$

Ecuación 2. Ganancia del circuito del sensor de tensión alterna.

El transformador TR1 posee dos circuitos primarios de 115V y dos circuitos secundarios de 5V, para su correcto funcionamiento se conectaron los dos circuitos primarios en serie y los circuitos secundarios en paralelo. El transformador utilizado además proporciona aislamiento galvánico, lo que proporciona mayor protección al microprocesador ante posibles cortocircuitos. El esquema del circuito está representado en la Figura 18. Justo después del transformador se clocó el diodo zener D29 cuya función principal es proteger el resto del circuito ante sobretensiones transitorias. El amplificador operacional U6A amplifica la tensión de entrada y le suma 2.5V. La señal de salida del amplificador U6A se envía al circuito de detección por cruce por cero. A continuación se añadió un divisor de tensión para generar la tensión AC Voltage y también se añadió el diodo doble D28, cuya función es mantener la tensión entre 0 y 3.3V.

El circuito de cruce por cero posee un comparador, este verifica continuamente si la tensión es mayor o menor de 2.5V. Cuando la tensión pase a ser mayor de 2.5V se activa el transistor Q1 mandando una señal al microprocesador.

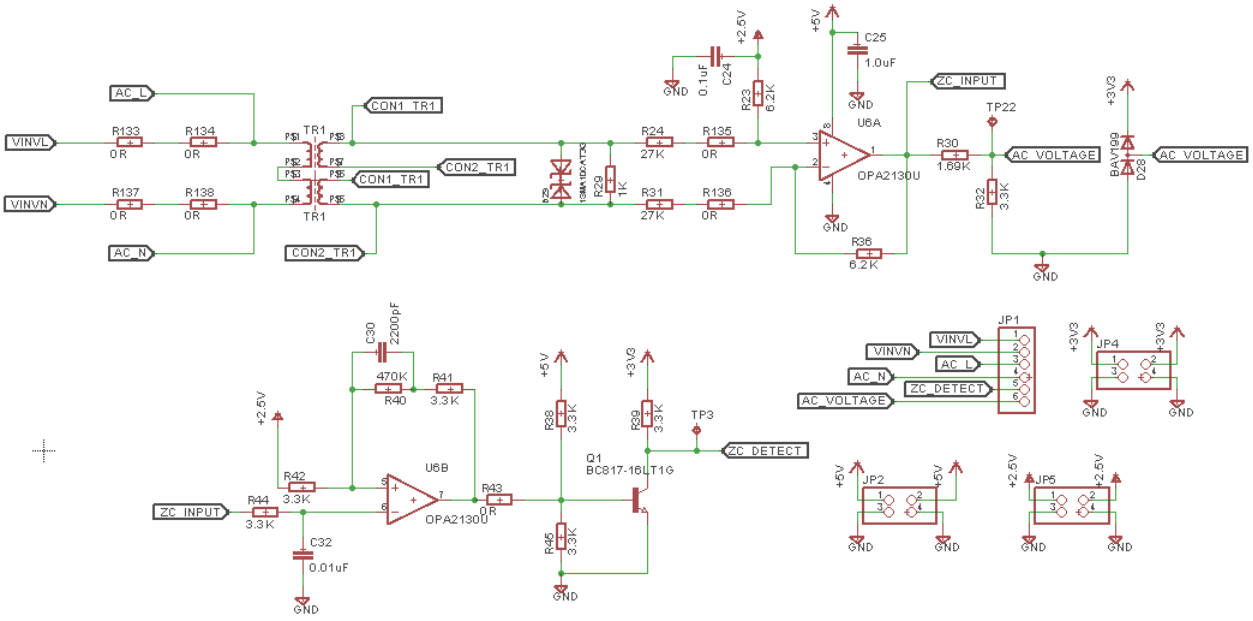

Figura 18. Esquema del sensor de tensión alterna.

A continuación se muestra el circuito impreso diseñado, que está representado en la Figura 19.

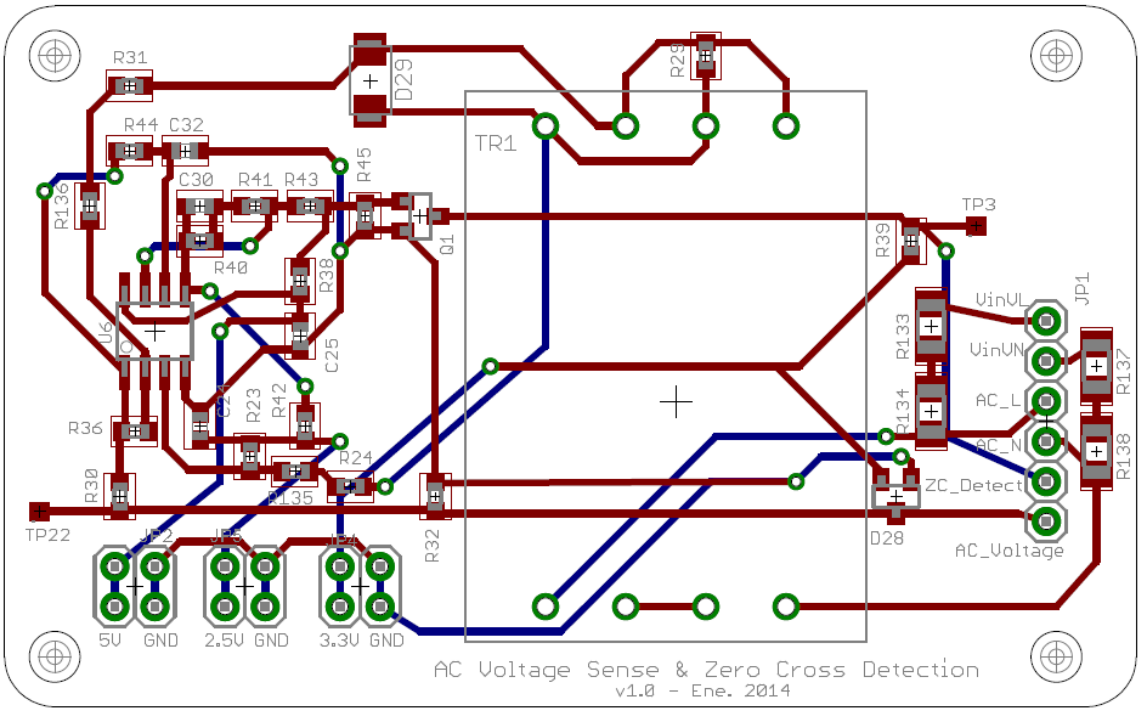

Figura 19. Circuito impreso del sensor de tensión alterna.

3.5 SENSOR DE CORRIENTE DEL FLYBACK

En esta sección se describe el circuito encargado de medir la corriente de los transformadores de los convertidores DC/DC. Las señales de entrada provienen directamente de los dos transformadores de corriente, que están incluidos en el circuito del convertidor Flyback. En primer lugar la corriente de salida de los transformadores de intensidad se convierte en una señal de tensión mediante una resistencia de 15 ohmnios.

Esta señal pasa por un circuito amplificador no inversor de ganancia 3.5 aproximadamente, de esta manera se ajusta el rango de la tensión a los valores del microprocesador. Seguidamente se incluye un condensador para estabilizar la tensión y un diodo doble para mantener la señal entre 0 y 3.3V. Posteriormente hay un divisor de tensión del que se obtiene una señal que se utiliza para detener el módulo PWM si hay sobre corriente en alguno de los Flyback. Esta señal es enviada al comparador analógico que posee el microprocesador, para que este la compare con una referencia fijada por software de control.

Por lo tanto este circuito genera dos salidas por cada transformador, una para obtener el valor de la corriente y otra para indicar al microprocesador que hay una sobre corriente. Para ver las conexiones véase la Figura 20.

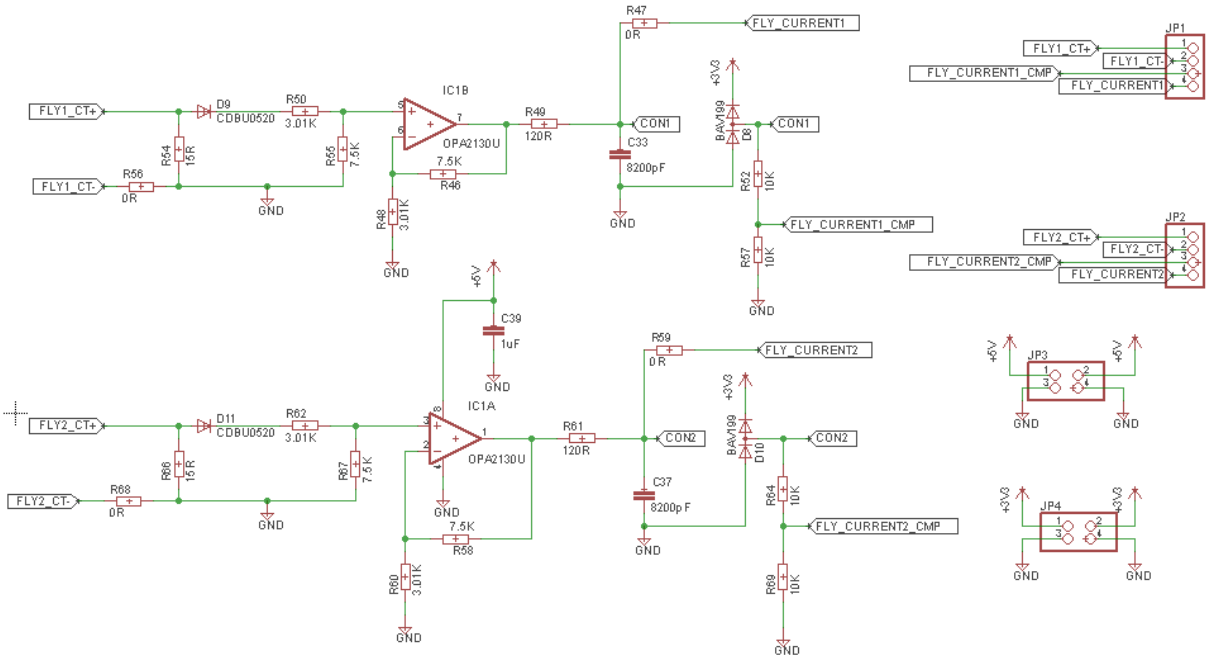

Figura 20. Esquema del sensor de corriente del Flyback.

A continuación se muestra el circuito impreso diseñado, que está representado en la Figura 21.

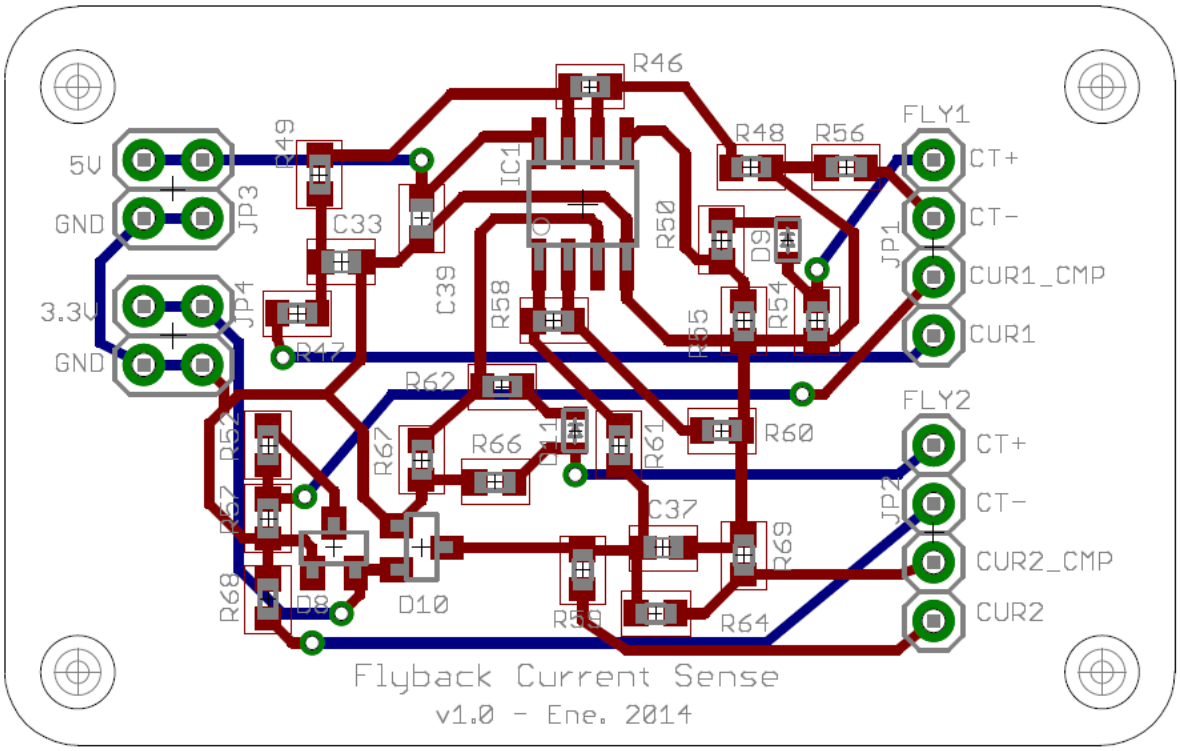

Figura 21. Circuito impreso del sensor de corriente del Flyback.

3.6 CONVERTIDOR FLYBACK

En esta sección se comentan los componentes del circuito que incluye los dos convertidores Flyback que funcionan conectados en paralelo. Son unos convertidores DC/DC encargados de elevar la tensión desde los valores de entrada, entre 20 y 45V, hasta el valor de la tensión de la red de distribución.

La tensión de entrada pasa al transformador a través de varios condensadores en paralelo, estos condensadores son de desacoplamiento. Su función es mantener la tensión aproximadamente constante incluso cuando la tensión de entrada sufra variaciones transitorias. Los transformadores TR4 y TR6 son los encargados de elevar la tensión y transmitir la potencia al módulo del inversor Full Bridge. En serie con dichos transformadores de tensión se situan los transformadores de intensidad TR5 y TR7. Estos transformadores generan una intensidad proporcional a la del circuito de potencia que se envía al circuito del sensor de corriente del Flyback.

Los transformadores TR4 y TR6, que son los encargados de elevar la tensión, están controlados por los transistores (Q6 Q7 Q8 y Q9), véase la Figura 22. Uno de los problemas que pueden surgir es que al apagar el transistor la energía almacenada en el núcleo magnético puede provocar sobre tensiones, para evitarlo se introdujeron algunos condensadores.

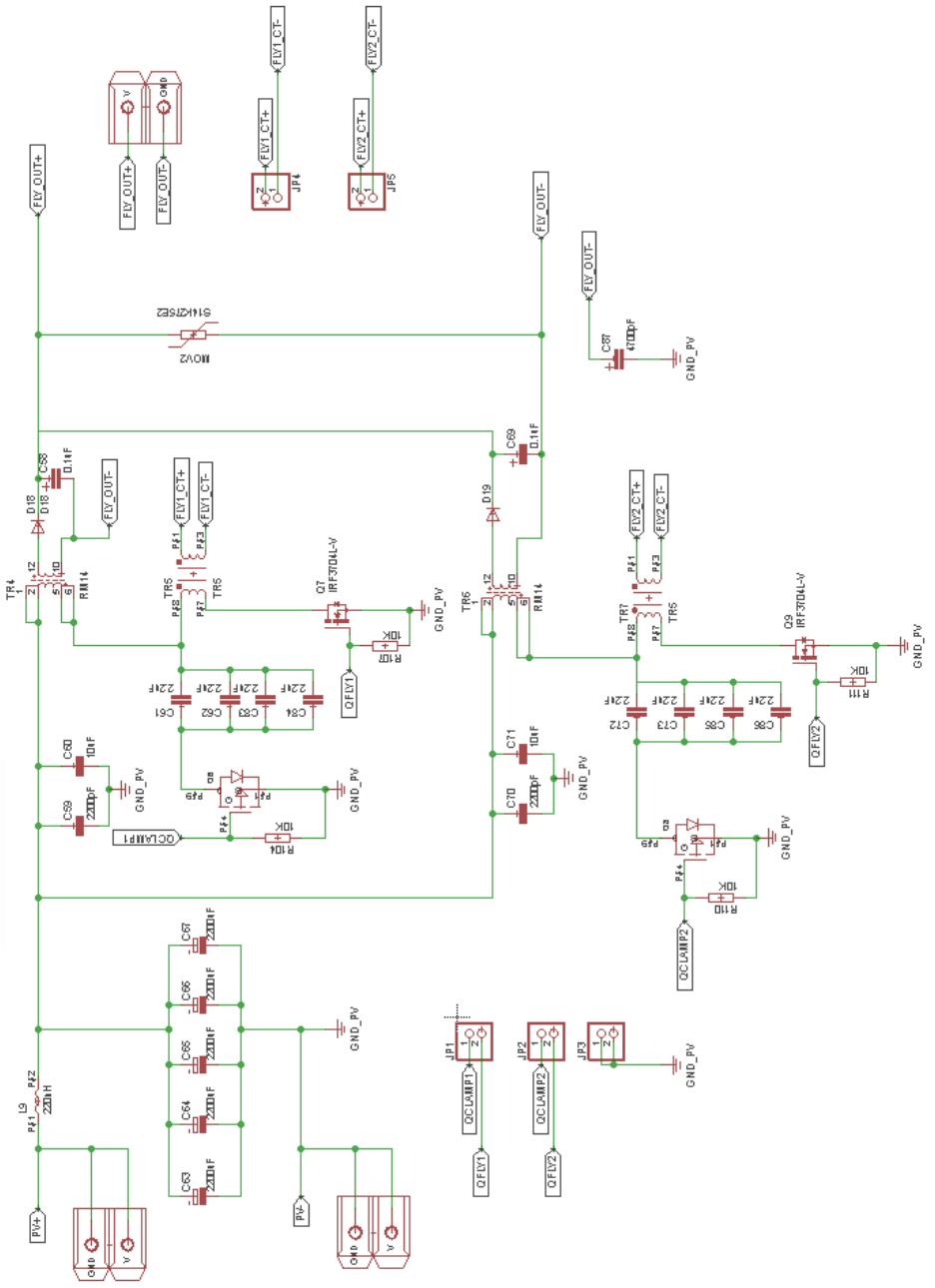

Figura 22. Esquema del convertidor Flyback.

En el circuito secundario de los transformadores TR4 y TR6 se encuentran los diodos D18 y D19. Dichos diodos se incluyeron en el diseño para asegurarse que la intensidad circulase del circuito primario de los transformadores al circuito secundario de los mismos. Por tanto el flujo de potencia siempre se producirá desde la entrada a la salida del convertidor Flyback. Tambien se añadieron los condensadores C58 y C69 en el circuito secundario de los transformadores para estabilizar la tensión.

Se instalan dos transformadores en paralelo para que compartan la corriente. Al compartir la corriente, se reducen las pérdidas en los conductores y en los transformadores. Además al ser más pequeña la corriente (aproximadamente un 50%), se reduce la oscilación (*ripple*) de la señal final mejorando la distorsión total armónica (*THD*). Por tanto el circuito primario y el circuito secundario de los dos transformadores de tensión están conectados en paralelo, tal y como se puede apreciar en la Figura 22.

A continuación se muestra el circuito impreso diseñado, que está representado en la Figura 23. En este circuito impreso es esencial verificar el grosor de las pistas que conectan los componentes del circuito, porque al tener bajas tensiones de entrada se tienen elevadas corrientes. El grosor de las pistas a pesar de aumentar el coste de fabricación es una característica necesaria para evitar el sobrecalentamiento del conjunto.

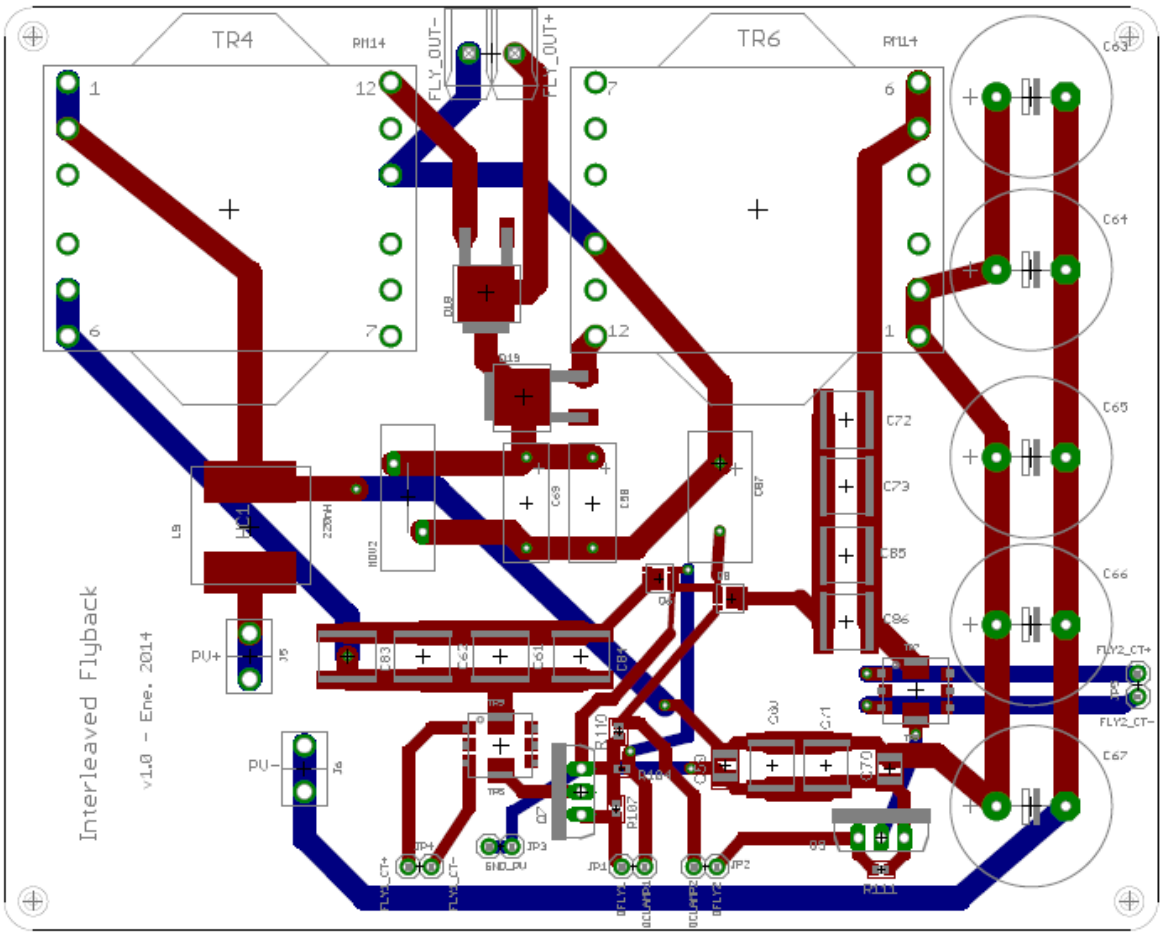

Figura 23. Circuito impreso del convertidor Flyback.

3.7 CONTROLADOR DEL CONVERTIDOR FLYBACK

En esta sección se muestra la configuración del circuito encargado de generar las señales que controlan el convertidor Flyback a través de los transistores. Las señales de entrada de este circuito son las señales generadas por el PWM del microprocesador en función del código de control, véase la Figura 24.

Estas señales se introducen en los controladores *(Drivers)* mediante un divisor de tensión y son estos los encargados de generar las señales que controlan los transistores. Para asegurarse que la tensión de alimentación de los controladores es constante hay dos condensadores en paralelo. A la salida de los controladores se encuentran cuatro diodos

de protección, uno por cada señal de control. Se incluyeron los diodos D22 y D25 para obligar a tener una tensión menor o igual a cero en las señales QCLAMP1 y QCLAMP2.

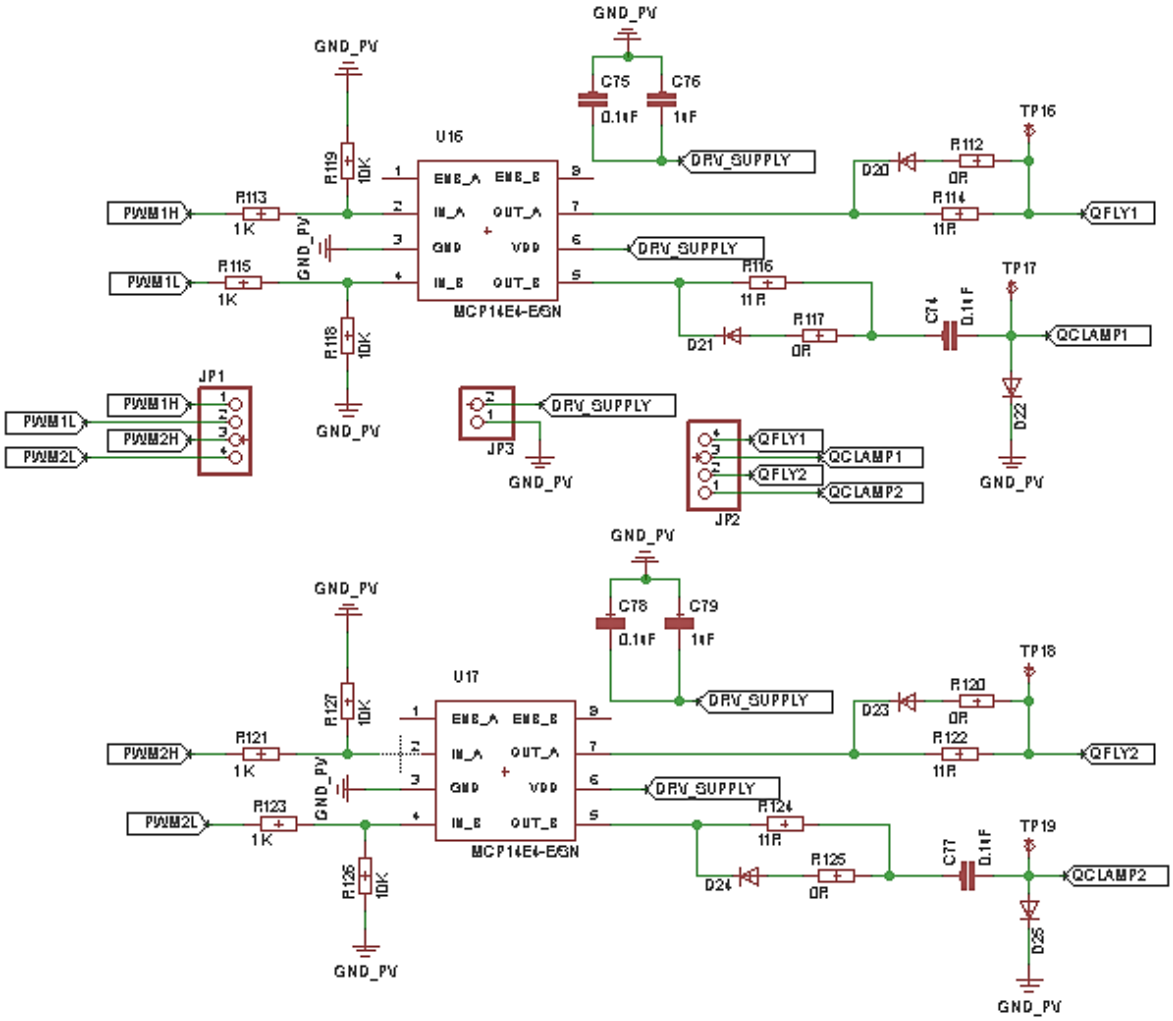

Figura 24. Esquema del controlador del Flyback.

Finalmente estas señales son las que llegan al convertidor Flyback comentado anteriormente. Las señales QCLAMP1 y QCLAMP2 se envían a los transistores Q6 y Q8 respectivamente. Mientras que las señales QFLY1 y QFLY2 se mandan a los transistores Q7 y Q9 respectivamente.

Seguidamente se muestra el circuito impreso diseñado, que está representado en la Figura 25.

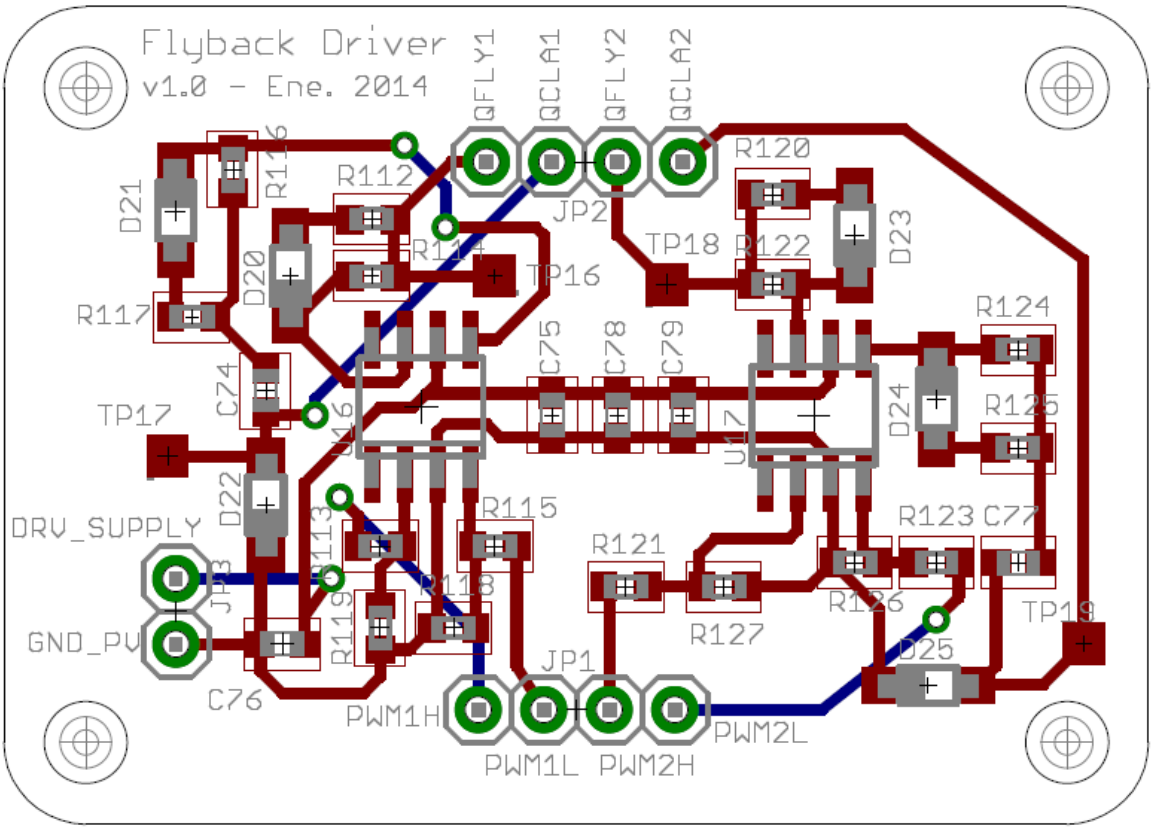

Figura 25. Circuito impreso del controlador del Flyback.

3.8 PUENTE H

En esta sección se describe el circuito que contiene el puente H (*Full Bridge*) es el encargado de generar una onda alterna a partir de una tensión continua.

La tensión de entrada de este circuito es la tensión que sale del convertidor Flyback DC/DC, esta tensión se transforma en una onda alterna mediante la actuación de los transistores de potencia. Las señales de control de dichos transistores provienen del circuito controlador que se explica en la sección 9 de este capítulo. Estas señales también se utilizan para que la onda alterna generada realice los cruces por cero al mismo tiempo que la tensión de referencia. Esta tensión de referencia puede ser la tensión de la red de distribución o una tensión teórica.

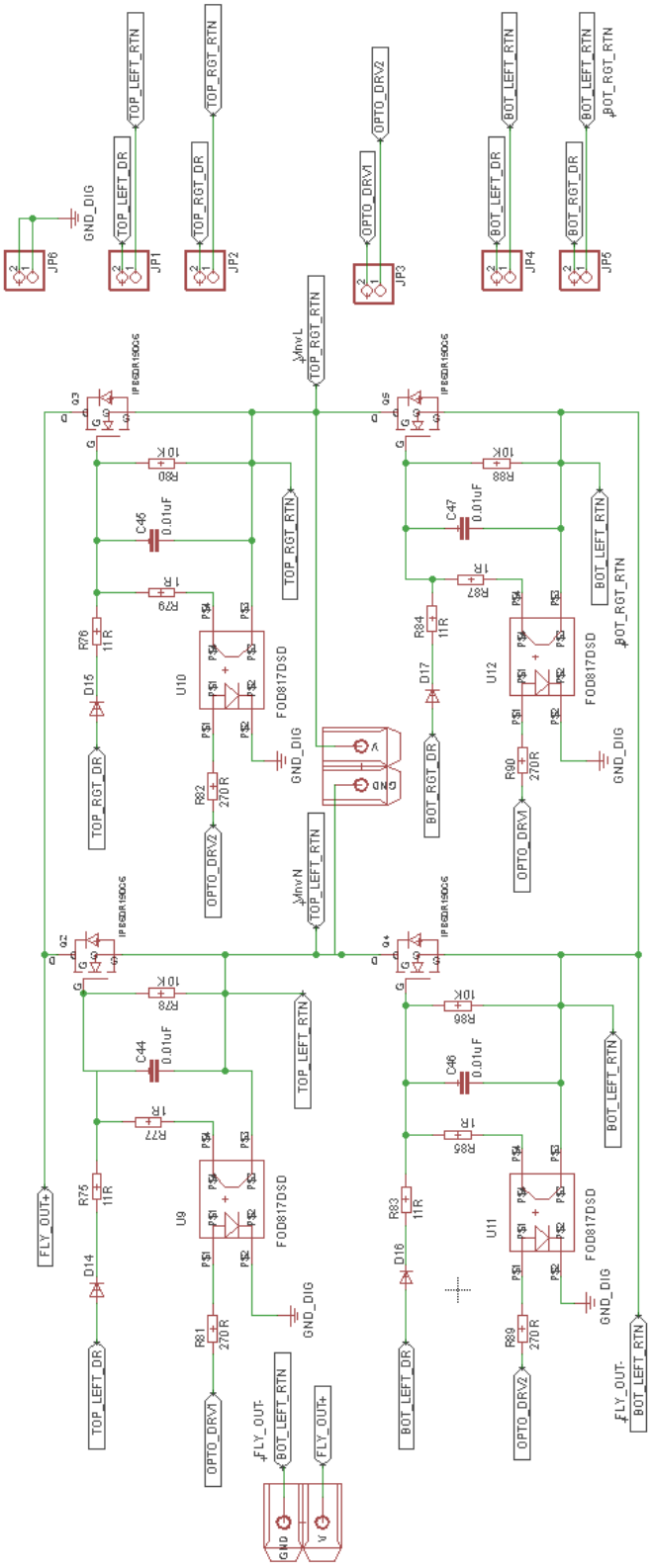

Figura 26. Esquema del puente H.

A continuación se muestra el circuito impreso diseñado, que está representado en la Figura 27.

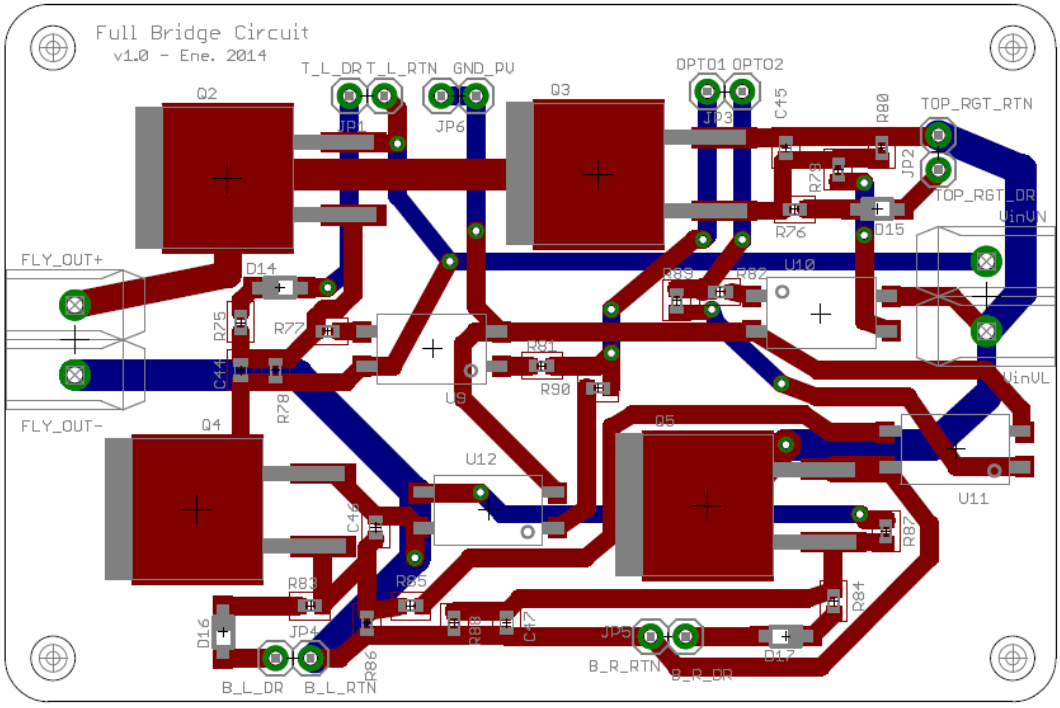

Figura 27. Circuito impreso del puente H.

3.9 CONTROLADOR DEL PUENTE H

En esta sección se muestra el circuito encargado de generar las señales de control del puente H y un diseño de circuito impreso del mismo. La señal de entrada de este circuito es la señal 3 del PWM del microprocesador. Esta señal se introduce en el controlador a través de un divisor de tensión, formado por una resistencia de 1 kohm y otra de 10 kohm por lo que la relación es prácticamente la unidad. Asimismo se incluyen dos condensadores en paralelo para estabilizar la tensión en el punto de entrada de la tensión de alimentación, denominada DRV_SUPPLY.

Se generan cuatro señales de control para los cuatro transistores que controlan el inversor, aunque en realidad las señales son iguales dos a dos. La señal de control del transistor de la zona izquierda superior es igual a la señal del transistor de la zona derecha inferior.

Igualmente la señal de control del transistor de la zona derecha superior es igual a la señal del transistor de la zona izquierda inferior.

Este duplicado de las señales de control se realiza mediante un transformador que posee un circuito primario y dos circuitos secundarios, véase la Figura 28. Además de duplicar la señal, el transformador proporciona aislamiento galvánico entre los componentes de control y los componentes que transmiten el flujo de potencia.

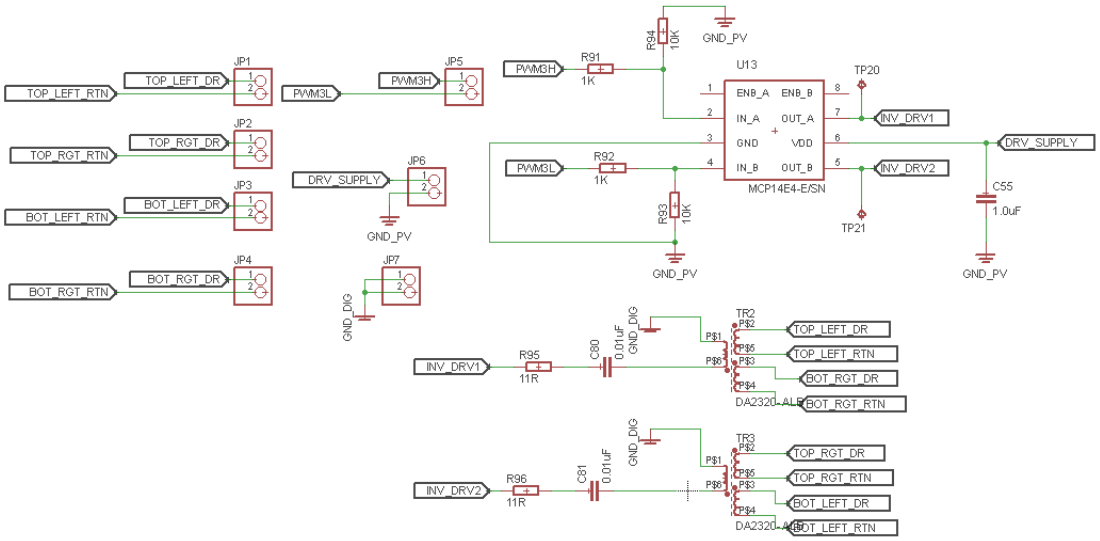

Figura 28. Esquema del controlador del puente H.

Seguidamente se muestra el circuito impreso diseñado, que está representado en la Figura 29. Se puede observar como en la parte superior se encuentran las conexiones para la señal de entrada. En la zona inferior están las conexiones de las cuatro señales de control producidas en el circuito.

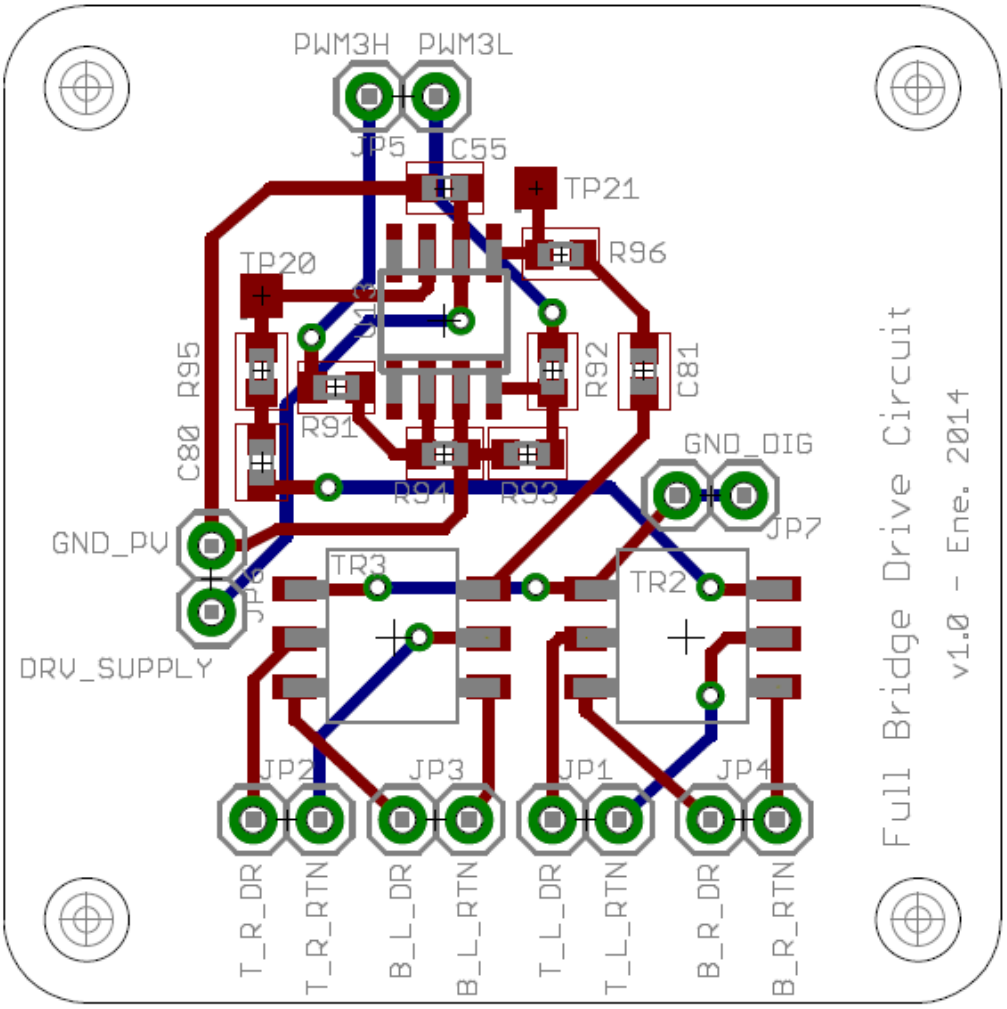

Figura 29. Circuito impreso del controlador del puente H.

3.10 PROTECCIÓN DE SOBRETENSIÓN DEL FLYBACK

En esta sección se describe el circuito de protección frente a sobre tensiones en el convertidor Flyback. La tensión que recibe es la de salida del convertidor Flyback y la tensión de salida sirve para mandar una señal al microprocesador.

La tensión recibida se transforma mediante un divisor de tensión antes de conectarla con el componente U15 mediante la Ecuación 3.

$$V_{FLY_OUT} \cdot \frac{R106}{R106 + R102 + R100 + R98} > 2.5V$$

Ecuación 3. Tensión que activa el circuito de protección del Flyback.

Para generar la señal FLY_VOLTAGE_CMP se usa el componente U15, un amplificador de error con aislamiento óptico. Si la tensión es suficientemente elevada se activará el transistor interno elevando la tensión de la señal de salida FLY_VOLTAGE_CMP, véase la Figura 30.

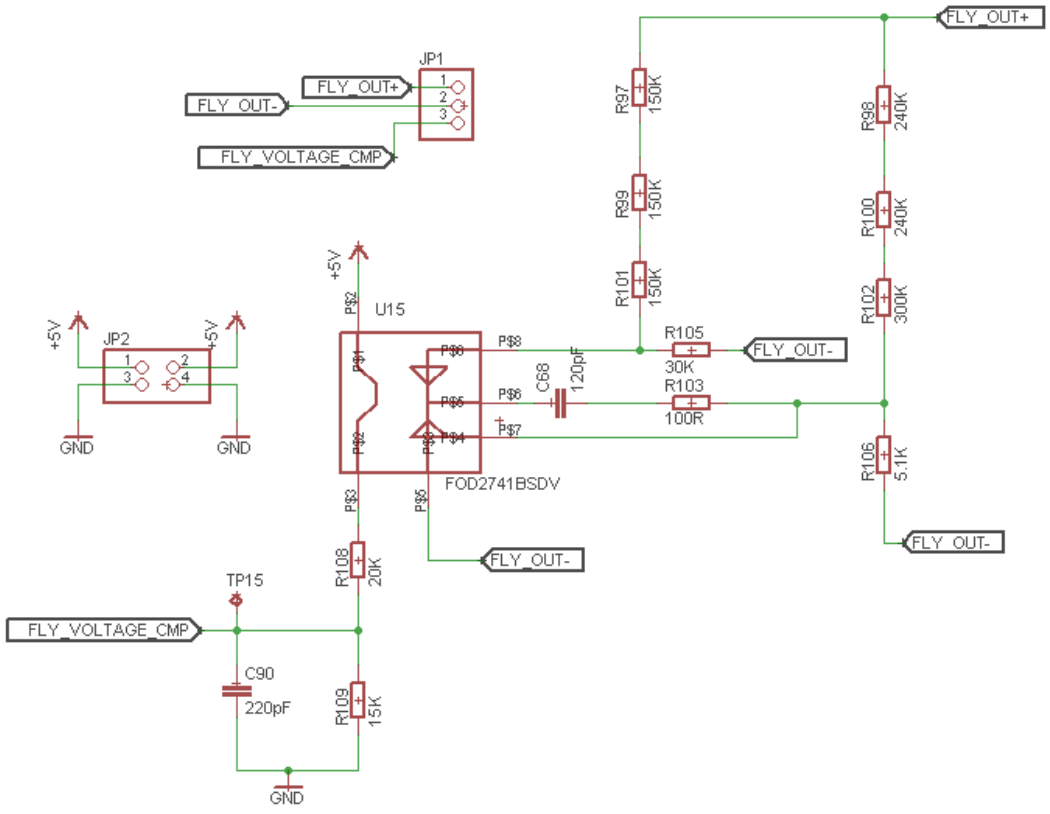

Figura 30. Esquema del circuito de sobretensión del Flyback.

Esta señal se envía al comparador analógico que posee el microprocesador, este componente comprueba si la señal es superior a un valor dado. Este valor de tensión analógica viene dado en función del valor del registro correspondiente. El valor del registro puede ser desde 0 hasta 1024 y se corresponde con un rango de tensiones entre 0 y 1.65V.

Esta señal permanecerá inactiva hasta que se produzca una sobretensión, momento en el cual la señal será superior al valor con el que se compara y se ejecutará una interrupción. Al ejecutar dicha interrupción el microprocesador apagará el inversor.

A continuación se muestra el circuito impreso diseñado, que está representado en la Figura 31.

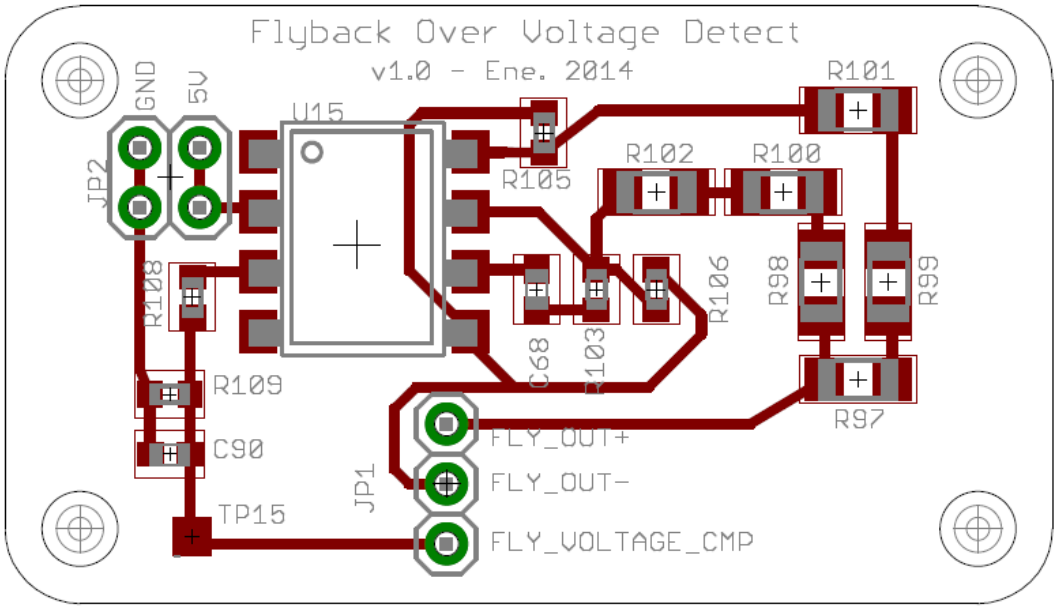

Figura 31. Circuito impreso de protección contra sobretensión en el Flyback.

3.11 CONEXIÓN CON EL MICROPROCESADOR

En esta sección se muestra un módulo que contiene varios circuitos de acondicionamiento y el circuito de conexión del microprocesador con todas las señales necesarias para su correcto funcionamiento. A continuación se analizarán de manera superficial los circuitos incluidos en este módulo.

En la Figura 32 se puede observar que señales se han conectado a cada uno de los pines del microprocesador. Destaca la señal de referencia de 2.5 V que en lugar de medirse directamente se mide a través de un divisor de tensión que contiene un condensador para estabilizar la tensión.

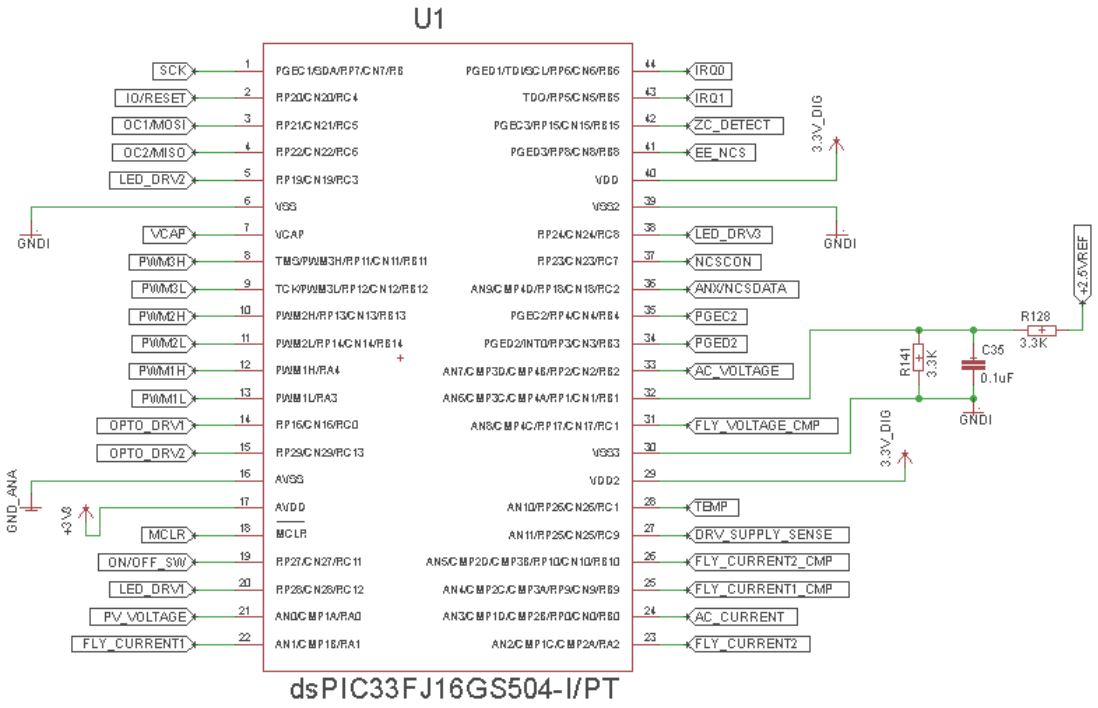

Figura 32. Esquema del circuito de conexión con el microprocesador.

En la Figura 33, se muestra el esquema del circuito de acondicionamiento que genera la tensión de alimentación a 12 V y la señal de medida de dicha tensión. La tensión de alimentación se obtiene a partir de la tensión continua de entrada mediante un regulador de tensión. La medida de dicha tensión se obtiene a través de un divisor de tensión.

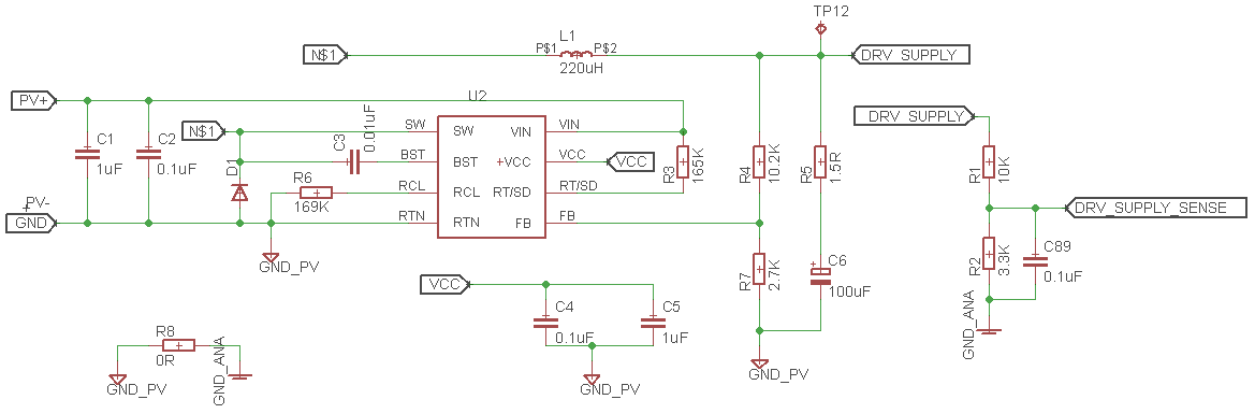

Figura 33. Circuito de acondicionamiento para la alimentación a 12 V.

En la Figura 34 se muestra el esquema del circuito de acondicionamiento que genera la tensión de alimentación a 5 V. Esta tensión se genera a partir de la tensión de 12V

mediante otro regulador. Destaca la presencia de varios condensadores tanto a la entrada como a la salida, su objetivo es estabilizar la tensión.

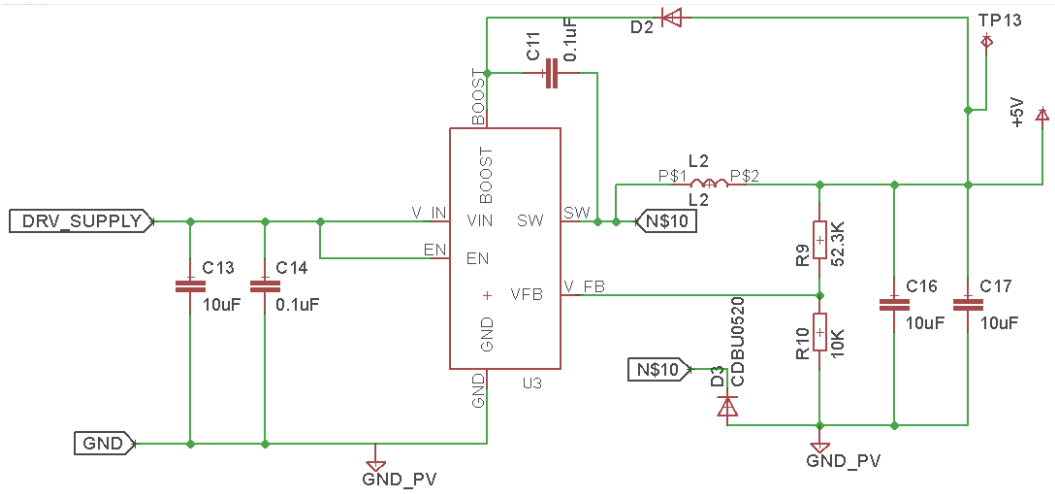

Figura 34. Circuito de acondicionamiento para la alimentación a 5 V.

En la Figura 35 se muestra el esquema del circuito de acondicionamiento que genera la tensión de alimentación a 3.3 V. Esta tensión se genera a partir de la tensión de 5 V mediante otro regulador. En este caso también hay varios condensadores para estabilizar las tensiones.

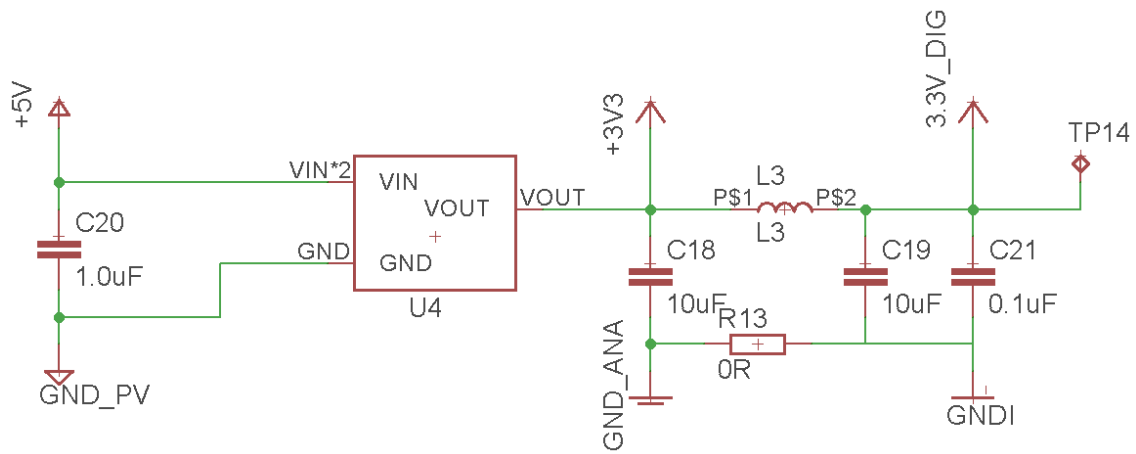

Figura 35. Circuito de acondicionamiento para la alimentación a 3.3 V.

En la Figura 36 se muestra el circuito de acondicionamiento encargado de encender los LEDs y comprobar el estado del interruptor general. El LED D27 es el encargado de indicar si se ha producido algún error. El LED D6 se enciende si el interruptor está en la posición de OFF. El LED D5 se utiliza para indicar que el microinversor está alimentado, en este inversor la tensión de alimentación proviene de la tensión continua de entrada.

El resto de los LEDs se han empleado para depurar el código durante el desarrollo del mismo y en la versión final no tienen un uso específico. Se controlan mediante señales digitales del microprocesador.

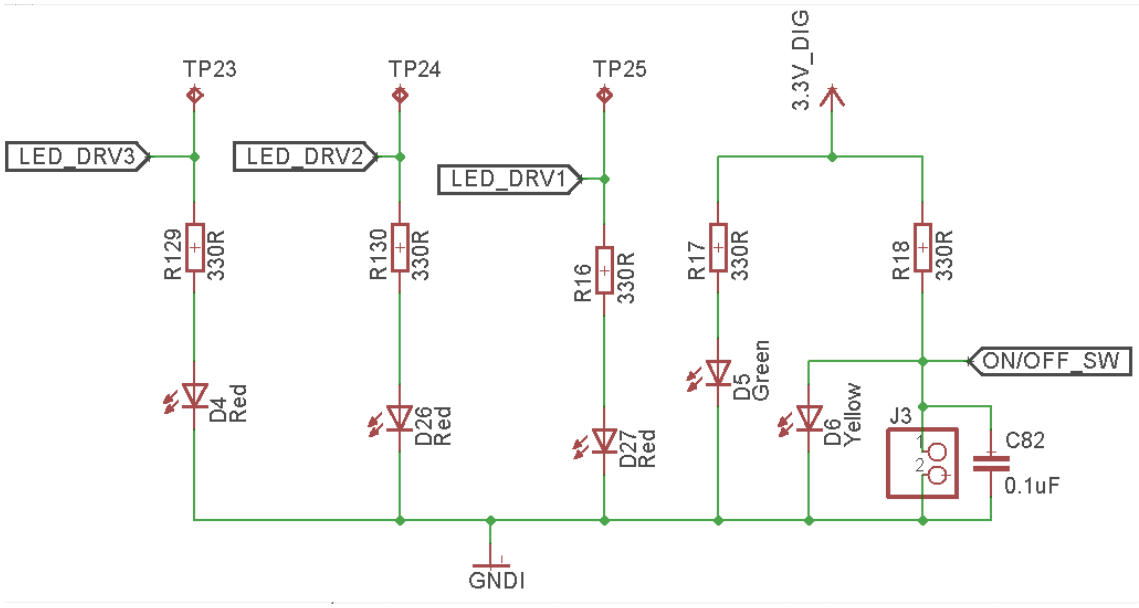

Figura 36. Circuito de acondicionamiento para los LEDs y el interruptor.

En la Figura 37 se muestra el circuito que acondiciona las señales que provienen del ordenador. Su función principal es transmitir los datos para programar el microprocesador.

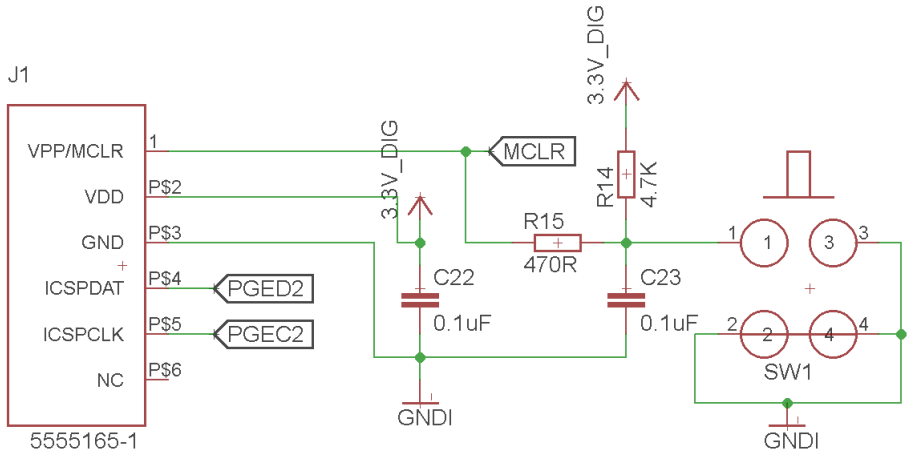

Figura 37. Circuito para la comunicación con el PC.

A continuación se muestra el circuito impreso diseñado, que está representado en la Figura 38. Este circuito impreso contiene todos los circuitos mencionados anteriormente, siendo U1 el microprocesador que controla el inversor.

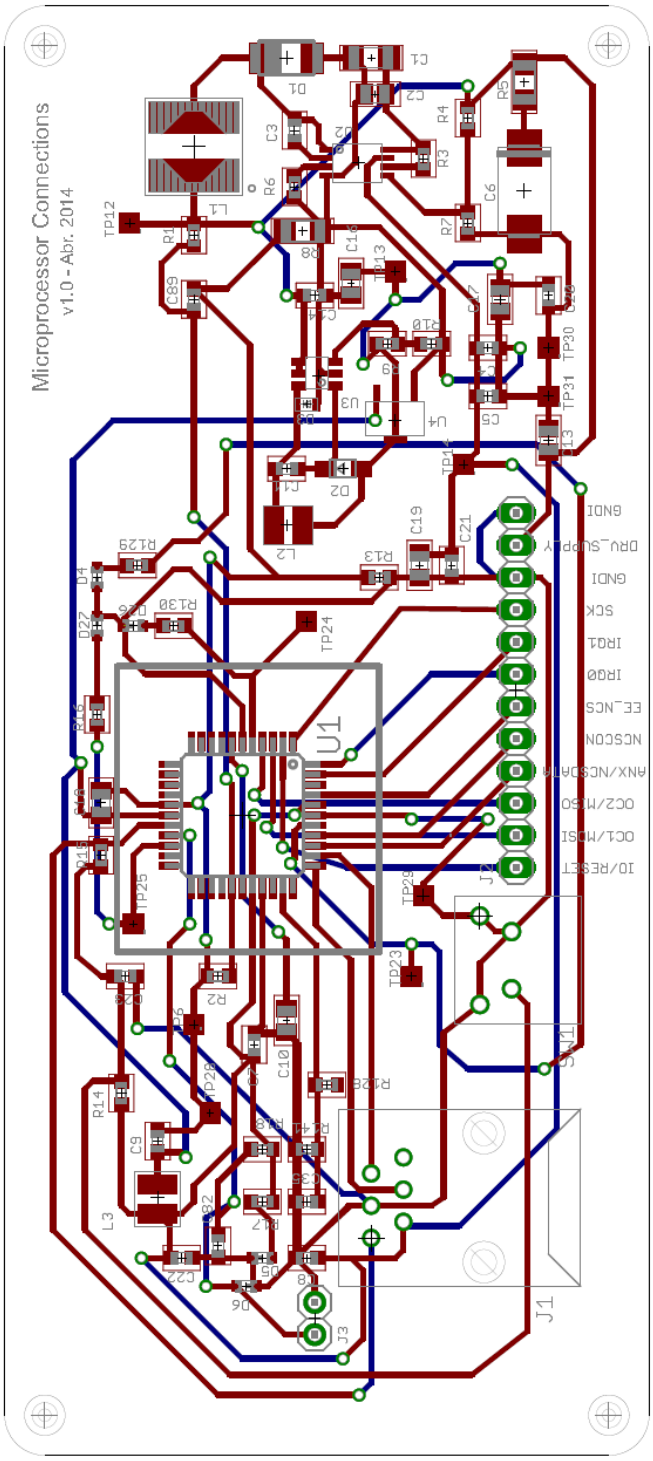

Figura 38. Circuito impreso de conexión con el microprocesador.

3.12 FILTRO EMI

En esta sección se indica el circuito del filtro EMI (*Electromagnetic interference filter*). La tensión de entrada de este circuito proviene del puente H, esta tensión ya es alterna pero contiene muchos armónicos de alta frecuencia. El objetivo de este circuito es atenuar todos estos armónicos superiores reduciendo así la distorsión total armónica (*THD*).

El filtro consiste en una bobina en modo común y un filtro en modo diferencial, véase la Figura 39. La salida del filtro es un varistor conectado entre la fase y el neutro, tras él se encuentran dos fusibles de protección.

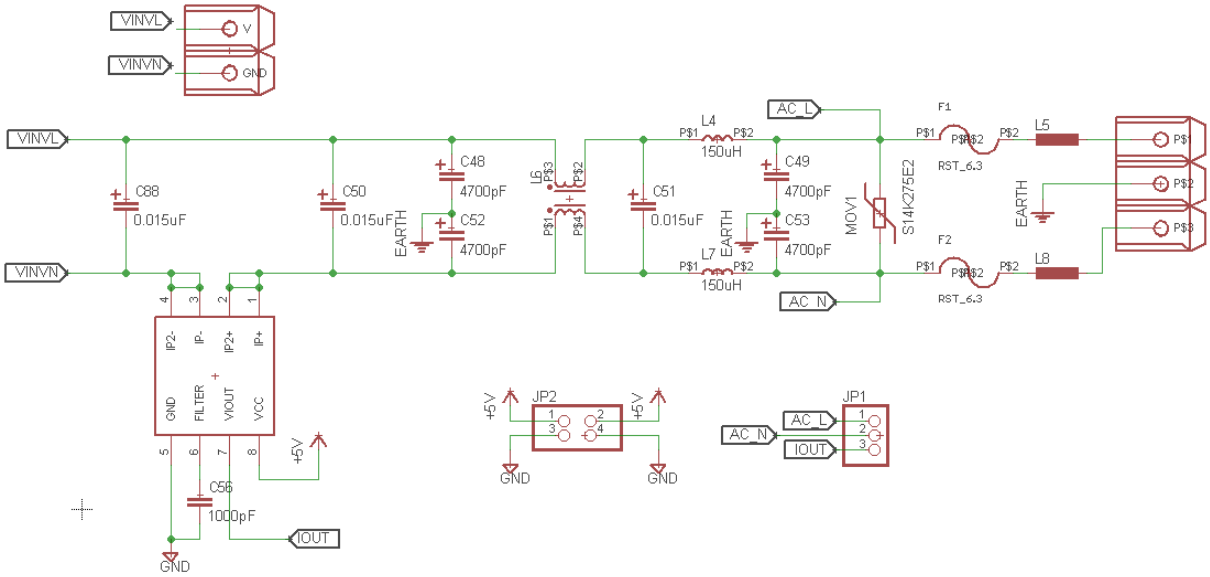

Figura 39. Esquema del filtro EMI.

Este circuito genera un nivel de tensión proporcional a la corriente de salida mediante un sensor. Esta medida junto con la diferencia de tensión medida en el varistor se usará para controlar el funcionamiento del inversor.

A continuación se muestra el circuito impreso diseñado, que está representado en la Figura 40.

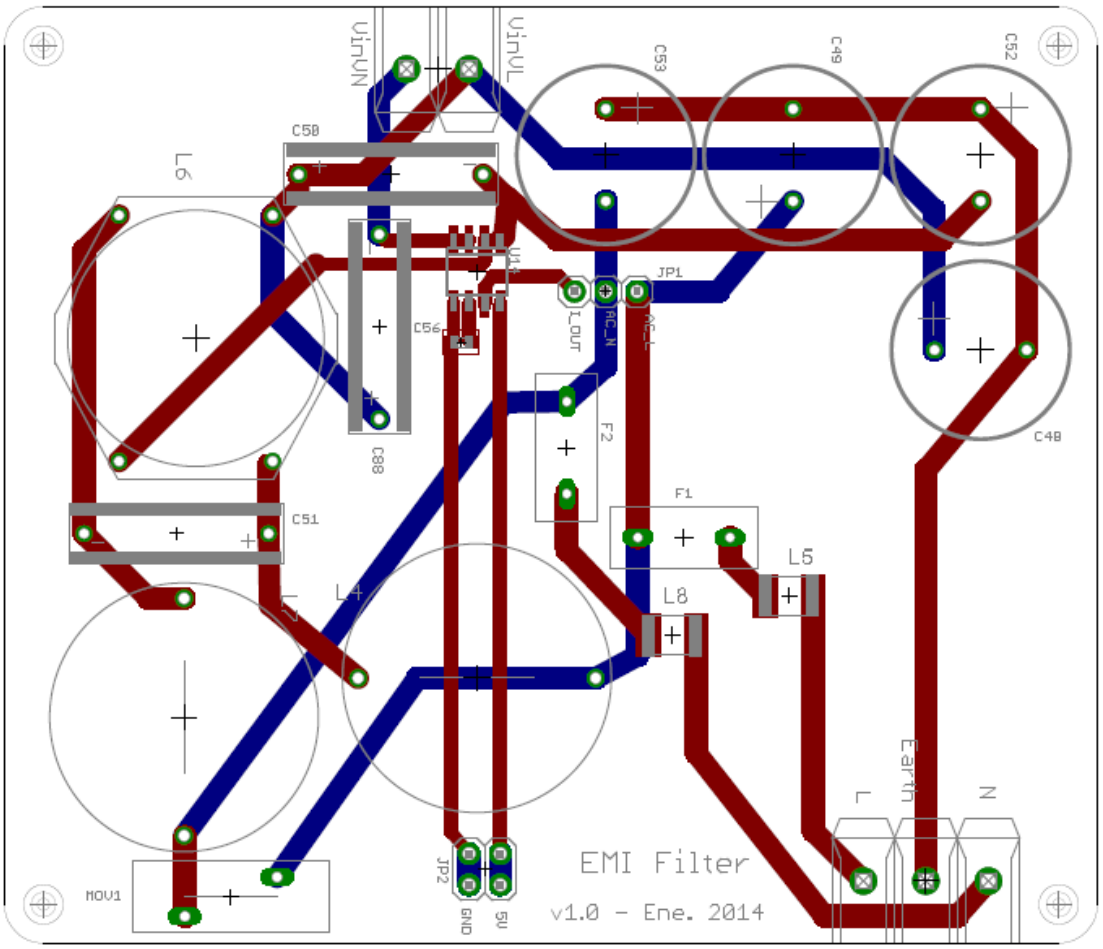

Figura 40. Circuito impreso del filtro EMI.

Capítulo 4 DESARROLLO DEL SOFTWARE

4.1 INTRODUCCIÓN

En este capítulo se describe el funcionamiento del programa en C que controla el microinversor AN1444. Este código está configurado para actuar de dos formas distintas en función de si existe o no una referencia de tensión alterna.

4.2 FUNCIONAMIENTO CONECTADO A LA RED DE DISTRIBUCIÓN

En esta sección se explica cómo actúa el código cuando tiene una tensión alterna de referencia, es decir, cuando el inversor está conectado a la red de distribución.

En primer lugar hay que tener en cuenta que el código se ha desarrollado a partir del código original del microinversor AN1444. Este código solo contemplaba la posibilidad de funcionar conectado a la red de distribución eléctrica. Su funcionamiento se basaba en la máquina de estados representada en la Figura 41. En esta máquina de estados hay tres estados: System Error, System Start Up y Day Mode. A continuación se describen estos tres estados.

En el estado System Error el inversor está apagado y no entrega potencia a la red. El inversor llegará a dicho estado cuando se produzca algún error o cuando el interruptor general este en la posición OFF.

Si ocurre algún error, en cualquier estado distinto de System Error, la variable fault pasará a tener un valor distinto de cero. Cuando se llega al estado System Error porque ha ocurrido algún error se genera un parpadeo en D27 que indica el error, para más información véase el manual de usuario en la parte III de este documento.

En cualquier caso en este estado se comprueba, cada 100 microsegundos, si hay algún error y en qué posición está el interruptor general. Si no hay ningún error y el interruptor está en la posición de ON, se activa un contador. Cuando el tiempo del contador es superior o igual a un segundo se produce una transición hacia el estado System Start Up.

Si ocurre algún error o cambia la posición del interruptor mientras el contador está activo, el tiempo del contador vuelve a cero.

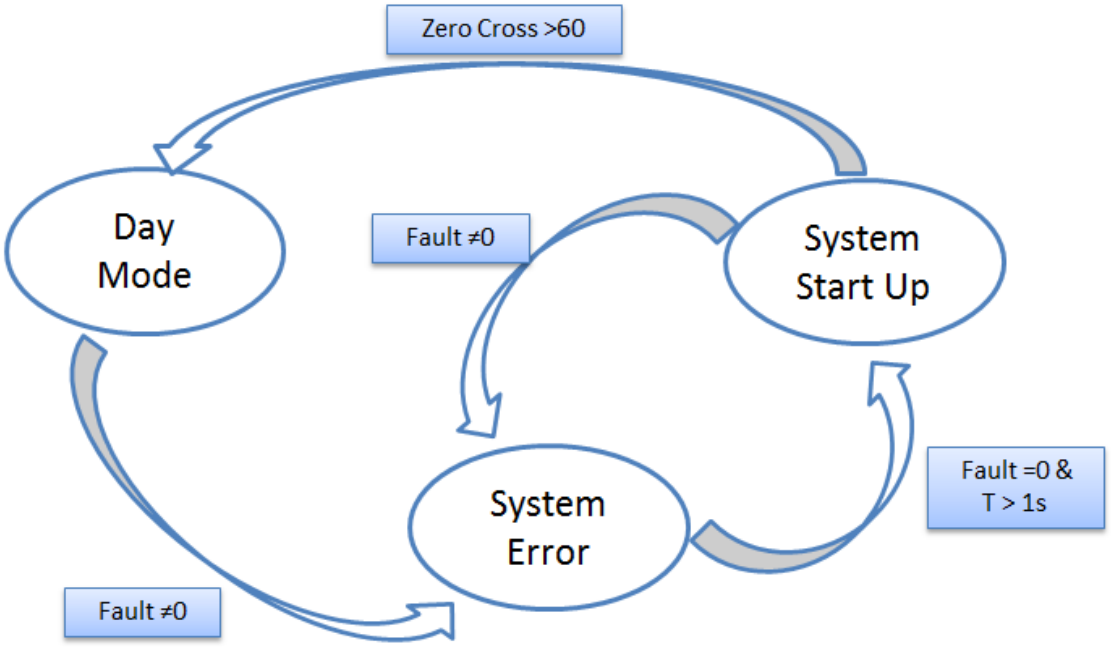

Figura 41. Máquina de estados del código original.

En el estado System Start Up todos los módulos empiezan a funcionar menos el inversor *Full Bridge.* En este estado se comprueba el offset de la corriente alterna y se calcula la frecuencia de la red de distribución (50 o 60 Hz). Al mismo tiempo se van contando los cruces por cero, cuando la tensión de referencia llega a 60 cruces por cero se activa el *Full Bridge* y se produce la transición al estado Day Mode. Las funciones principales de este estado son evitar arrancar el sistema con algún error y saber cuál es la frecuencia de la red de distribución eléctrica. Cada vez que se ejecuta la máquina de estados (100 us) se comprueba si ha ocurrido algún error o si el interruptor cambia a la posición OFF, si es así se produce una transición al estado System Error.

En el estado Day Mode el microinversor está completamente operativo y entrega potencia a la red eléctrica. En este estado se ejecutan los algoritmos de control que son tres: el MPPT, el control de los convertidores DC/DC y el control del puente H. Estos controles se explicarán más en detalle en este mismo capítulo. Cada vez que se ejecuta la máquina de estados (100 us) se comprueba si ha ocurrido algún error o si el interruptor cambia a la posición OFF, si es así se produce una transición al estado System Error.

Tomando como base la máquina de estados original, se desarrolló una nueva máquina de estados más compleja para poder incluir el funcionamiento en modo isla (desconectado de la red de distribución eléctrica). Dicha máquina de estados está recogida en la Figura 42.

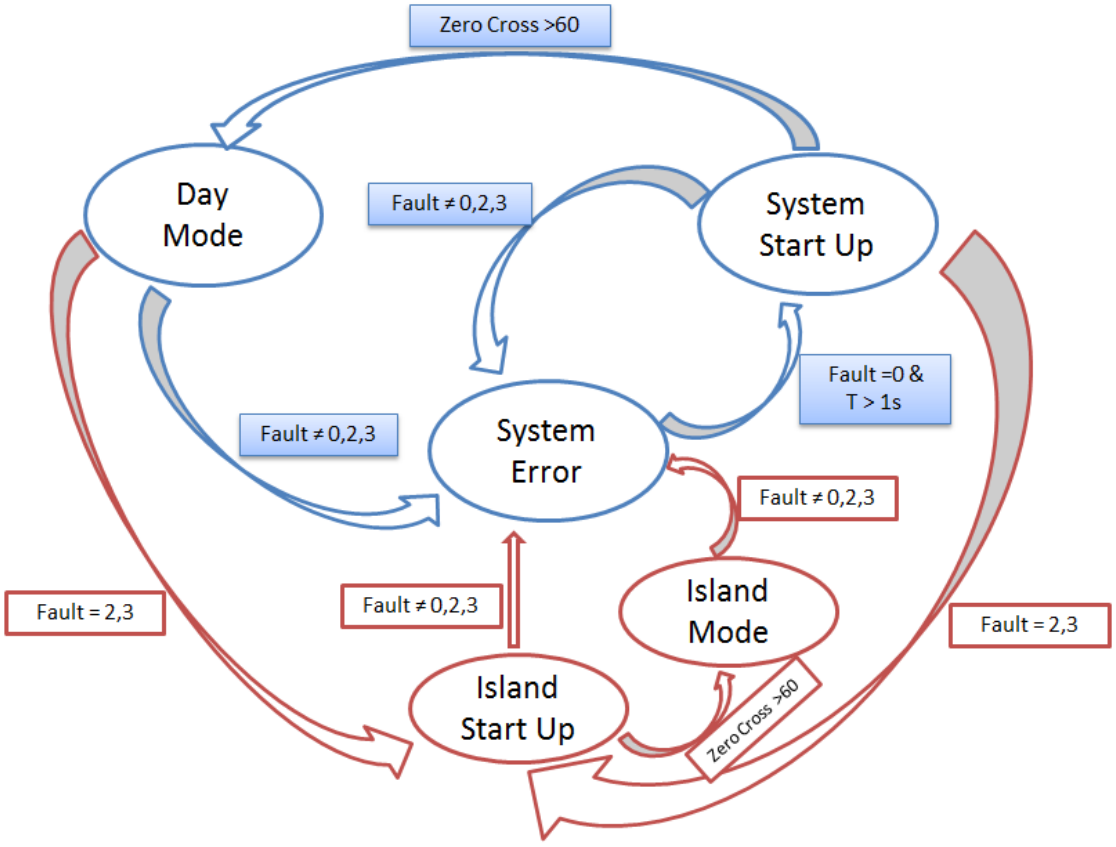

Figura 42. Máquina de estados del código desarrollado.

En la nueva máquina de estados se incluyeron dos estados más y se reprogramaron las transiciones entre los estados. Anteriormente no se distinguía entre unos tipos de errores y otros, pero se vio la necesidad de distinguir los errores relacionados con la onda alterna de referencia respecto del resto de los errores. El error número 2 indica que la frecuencia de la tensión está fuera de los límites establecidos y el error número 3 indica que el pico de la tensión es excesivamente bajo. Cuando alguno de estos dos errores existe significa que será necesario trabajar de manera aislada, ya que ha ocurrido algún problema con el suministro eléctrico.

El estado System Error funciona exactamente igual que en la máquina de estados anterior, pero en este caso solo los errores distintos del 2 y del 3 provocarán una transición a este estado.

En el estado System Start Up las tareas que se realizan son las mismas que en el diseño anterior exceptuando las transiciones entre estados. Cada vez que se ejecuta la máquina de estados (100 us) se comprueba si ha ocurrido algún error o si el interruptor cambia a la posición OFF. Si el interruptor está en la posición OFF o el error que ha ocurrido es distinto del 2 y del 3, se producirá una transición hacia el estado System Error. Si ha ocurrido el error 2 o el error 3, la transición que se realizará será hacia el estado Island Start Up.

El estado Day Mode funciona de la misma manera que lo hacía en el diseño anterior, exceptuando las transiciones entre estados. Cada vez que se ejecuta la máquina de estados (100 us) se comprueba si ha ocurrido algún error o si el interruptor cambia a la posición OFF. Si el interruptor está en la posición OFF o el error que ha ocurrido es distinto del 2 y del 3, se producirá una transición hacia el estado System Error. Si ha ocurrido el error 2 o el error 3, la transición que se realizará será hacia el estado Island Start Up.

En el estado Island Start Up empiezan a funcionar todos los módulos del microinversor, se comprueba el offset de la corriente alterna y se fija la frecuencia de funcionamiento de la microrred a 50 Hz. En este caso los cruces por cero no se miden mediante el hardware, sino que se cuentan los cruces por cero de una tensión teórica de referencia. Cuando se han producido 60 cruces por cero se produce la transición al estado Island Mode. Cada vez que se ejecuta la máquina de estados (100 us) se comprueba si ha ocurrido algún error o si el interruptor cambia a la posición OFF. Si el interruptor está en la posición OFF o el error que ha ocurrido es distinto del 2 y del 3, se producirá una transición hacia el estado System Error.

En el estado Island Mode se realizan las mismas tareas que en el estado Day Mode pero tomando como tensión de referencia la tensión generada. Esta tensión de referencia ficticia se genera en los estados Island Start Up e Island Mode. Los tres algoritmos de control que se ejecutan son los mismos pero cambian la referencia a seguir, se explicará más en detalle en la siguiente sección de este capítulo. Cada vez que se ejecuta la máquina de estados (100 us) se comprueba si ha ocurrido algún error o si el interruptor cambia a la posición OFF. Si el interruptor está en la posición OFF o el error que ha ocurrido es distinto del 2 y del 3, se producirá una transición hacia el estado System Error.

A continuación se comentará el funcionamiento de los controles que se llevan a cabo durante el funcionamiento cuando el microinversor está conectado a la red de

distribución. Para poder ver mejor los lazos de realimentación véase el diagrama de bloques recogido en la Figura 43. Se puede observar que hay tres controles superpuestos: el algoritmo MPPT, el control del balance de carga y el control del *Full Bridge*.

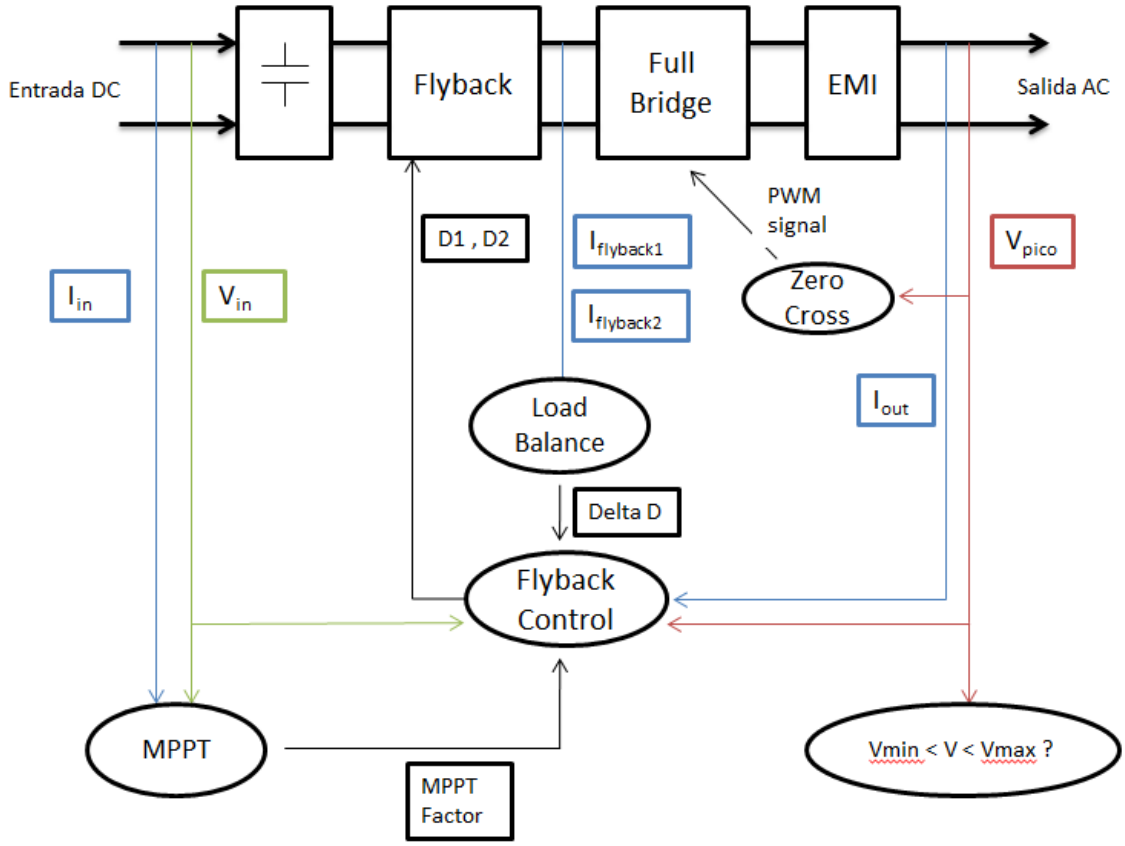

Figura 43. Diagrama de bloques del funcionamiento conectado a red.

El algoritmo MPPT es el encargado de controlar que la zona de tensión continua trabaje a la tensión óptima. Es decir, el valor de tensión a la entrada que maximiza la potencia entregada a la red eléctrica. La función del MPPT, teniendo en cuenta el nivel de tensión a la entrada y la corriente de entrada, genera un coeficiente que se envía a la función de control de los Flyback.

Este lazo de control se ve afectado por el lazo que controla el reparto equitativo de la carga entre los dos convertidores Flyback. Este segundo control mide la corriente de cada uno de ellos y calcula una variación del ciclo de trabajo (*duty cycle*). Es un control de tipo proporcional interal. Esta variación del ciclo de trabajo se suma al ciclo de trabajo de uno de los Flyback y se resta al ciclo de trabajo del otro Flyback.

Para unir estos dos controles está la función Flyback Control, que posee un control proporcional integral. Los datos de entrada de dicha función son: la tensión máxima a la salida, la corriente de salida, la tensión de entrada, el factor del algoritmo MPPT y la

variación de ciclos de trabajo del balance de carga. Con esta información la función genera los ciclos de trabajo de cada uno de los Flyback, estos valores digitales se comunican a los transistores de control mediante el PWM del microprocesador.

Por otro lado está el control del inversor *Full Bridge*, este control es el encargado de controlar el sincronismo entre la tensión de referencia y la tensión generada. Para ello toma los valores de pico y los cruces por cero de la tensión de referencia. Verifica que el pico de la onda generada coincida con el pico de la tensión de red y en caso contrario calcula las modificaciones necesarias. Este lazo de control actúa sobre el inversor *Full Bridge* mediante otro de los generadores PWM que posee el microprocesador.

4.3 FUNCIONAMIENTO EN MODO ISLA

En esta sección se explicará el funcionamiento de los controles cuando el microinversor trabaja aislado de la red de distribución eléctrica, es decir, en modo isla. Para una mayor comprensión del funcionamiento véase la Figura 44. La idea generales que en modo isla la tensión de referencia de los controles en vez de ser la tensión de salida es una tensión calculada de manera teórica. Para generar la variable tiempo de esta referencia teórica se incluyó una interrupción adicional, que se ejecuta cada 100 microsegundos.

El algoritmo MPPT es el encargado de controlar que la zona de tensión continua trabaje a la tensión óptima. La función del MPPT, teniendo en cuenta el nivel de tensión a la entrada y la corriente de entrada, genera un coeficiente que se envía a la función de control de los Flyback. Por este motivo la función de este control que no se ve afectada por el cambio de referencia.

El segundo control mide la corriente de cada uno de ellos y calcula una variación del ciclo de trabajo (*duty cycle*). Esta variación del ciclo de trabajo se suma al ciclo de trabajo de uno de los Flyback y se resta al ciclo de trabajo del otro Flyback. Por lo tanto esta función tampoco se ve afectada directamente por el cambio de la referencia.

Sin embargo la función de control de los Flyback si se ve afectada por este cambio de referencia. Esto se debe a que sus datos de entrada son: la tensión máxima de la referencia teórica, la corriente de salida, la tensión de entrada, el factor del algoritmo MPPT y la variación de ciclos de trabajo del balance de carga. Con esta información la función genera los ciclos de trabajo de cada uno de los Flyback, estos valores digitales se comunican a los transistores de control mediante el PWM del microprocesador.

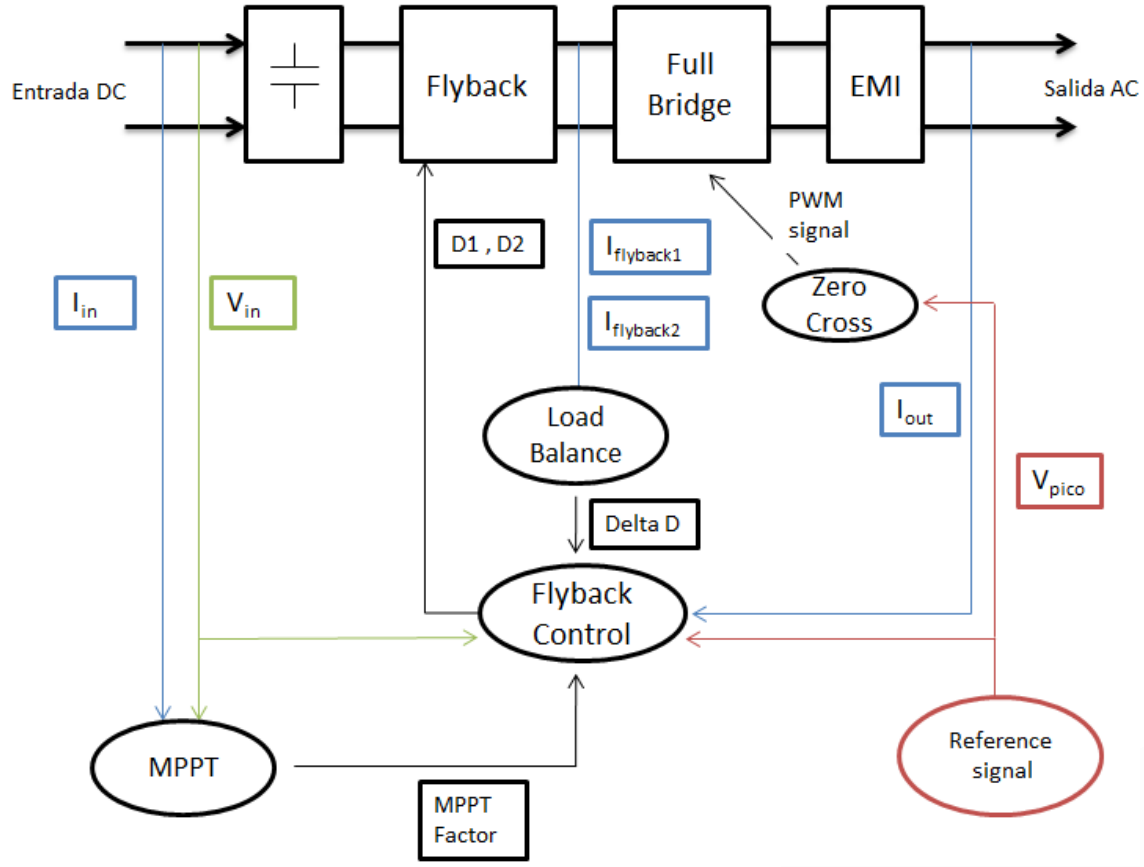

Figura 44. Diagrama de bloques del funcionamiento en modo isla.

Por último el control del inversor *Full Bridge* se ve notablemente afectado, porque es el encargado de controlar el sincronismo entre la tensión de referencia y la tensión generada. Para ello en esta ocasión toma los valores de pico y los cruces por cero de la tensión generada por el microprocesador, en vez de tomarlos de la tensión de la red eléctrica. Verifica que el pico de la onda generada coincida con el pico de la tensión de referencia y en caso contrario calcula las modificaciones necesarias. Este lazo de control actúa sobre el inversor *Full Bridge* mediante otro de los generadores PWM que posee el microprocesador.

Capítulo 5 RESULTADOS

En este capítulo se explican las pruebas realizadas y los resultados obtenidos del trabajo obtenido tras la ejecución del proyecto fin de carrera. Las pruebas que se realizaron fueron dos: la verificación del funcionamiento conectado a la red de distribución y la verificación del funcionamiento en modo isla.

En la primera de ellas se observó el adecuado funcionamiento del microinversor cuando ha de seguir la referencia de tensión marcada por la red de distribución eléctrica. La referencia se siguió de manera correcta. Posteriormente se calculó el rendimiento del inversor en diferentes puntos de funcionamiento calculando la potencia de entrada y de salida, véase la Figura 45.

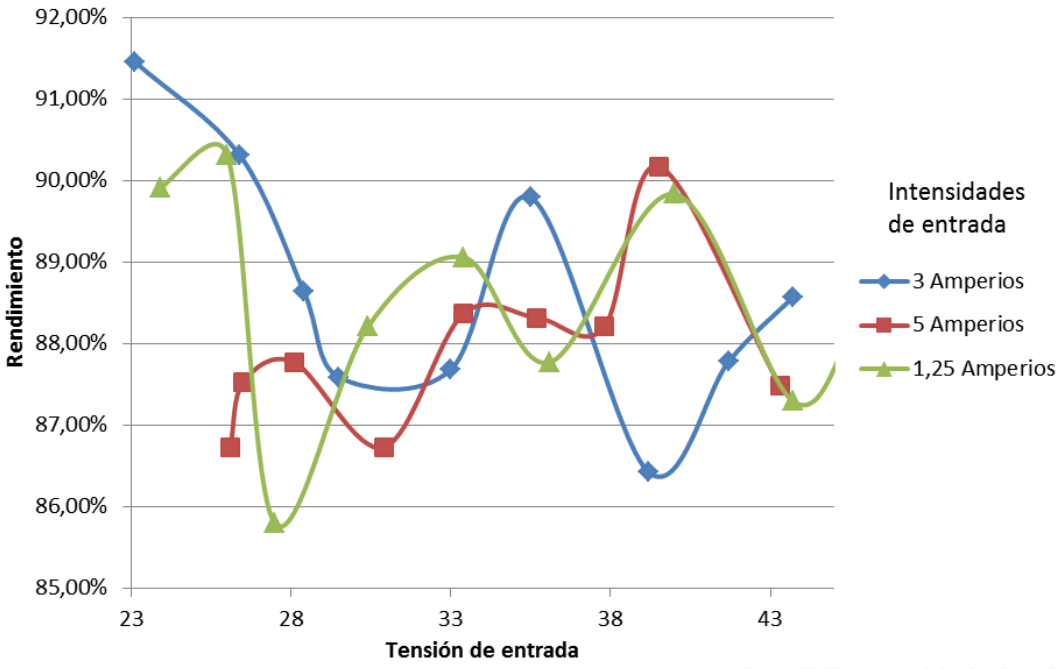

Figura 45. Rendimientos del microinversor conectado a la red de distribución.

La segunda prueba realizada en modo isla se observó el correcto funcionamiento del microinversor cuando ha de generar la referencia de tensión. La referencia se generó y se siguió de manera adecuada. La distorsión total armónica obtenida fue menor del 5%.

A continuación se exponen los resultados obtenidos:

- Se realizaron los diseños en Eagle de cada uno de los módulos, incluyendo diseños esquemáticos y diseños de circuito impreso.

- Se consiguió la fabricación del inversor a un coste inferior a 2000 € como se había fijado en los objetivos, el coste de fabricación final ascendió a 887.6 €.

- Se logró el correcto funcionamiento del microinversor conectado a la red de distribución.

- Se alcanzó el objetivo de que el microinversor funcionase en modo isla.

Capítulo 6 CONCLUSIONES

Como conclusión se ha conseguido desarrollar un inversor capaz de funcionar conectado a la red y en modo isla, con un coste inferior al presupuestado, por lo que se ha cumplido el objetivo principal del proyecto.

Este inversor cumple los objetivos del proyecto ya que: es reprogramable, tiene una potencia de 200 W, optimiza las condiciones de funcionamiento y su coste de fabricación final es 887.6 €.

Este proyecto ha desarrollado una base sólida para que, en un futuro cercano, se consiga desarrollar una microrred completa y eficiente. Mediante el desarrollo de las microrredes se conseguirá una reducción del consumo eléctrico, debido a su fácil integración con energías renovables.

Además al promover la generación distribuida se reducirán las pérdidas en las redes de transporte. Esta reducción del consumo y de las pérdidas de energía permitirá el desarrollo de un futuro más sostenible y ecológico para todos.

Durante la elaboración del proyecto se ha tenido que hacer frente a numerosos problemas y contratiempos. Dos de ellos han supuesto un retraso considerable en el avance del proyecto. Es importante se tengan presentes en futuros proyectos y desarrollos similares.

El primero de ellos fue encontrar y recibir todos los componentes necesarios para la fabricación de la tarjeta. Esto se debió a que al tener más de 150 elementos fue necesario pedir los componentes a varios distribuidores. El problema se resolvió mediante recuentos exhaustivos de la lista de materiales para verificar que componentes faltaban por pedir y cuáles no.

El segundo fue que durante la verificación de su correcto funcionamiento se produjo un corto circuito. Por lo que fue necesario volver a fabricar el inversor. Este cortocircuito se produjo al depurar el código mientras funcionaba conectado a la red de distribución, ya que los mecanismos de protección no actuaron correctamente. Finalmente para evitar que se volviese a dar una situación parecida, no se volvió a depurar el código mientras el inversor estaba conectado a la red de distribución eléctrica.

Capítulo 7 FUTUROS DESARROLLOS

El desarrollo futuro más inmediato sería realizar una microrred con el inversor diseñado. Tal y como se comentó en el capítulo 1 los inversores son un componente fundamental en las microrredes y es por ello que uno de los desarrollos más interesantes sería desarrollar una microrred. Para que dicha microrred pudiese funcionar correctamente se necesitarían varios inversores, uno para cada fuente de energía que trabaje con tensión continua. Si la microrred estuviese conectada a la red de distribución todos los inversores funcionarían siguiendo la referencia de la red. Si, por el contrario, la microrred estuviese aislada lo más sencillo sería usar la configuración maestro esclavo. En esta configuración un inversor trabaja en modo isla y los demás seguirían la referencia marcada por este.

Por otro lado se podría desarrollar el hardware y el software necesario para la reconexión con la red de distribución. Consistiría en que una vez que la red de distribución eléctrica vuelva a estar operativa, el microinversor vuelva a los estados de funcionamiento estándar sin necesidad de apagarlo.

BIBLIOGRAFÍA

[1] S. M. Tennakoon, W. W. L. Keerthipala, and W. B. Lawrance, "Solar energy for development of a cost-effective building energy system," in International Conference on Power System Technology, 2000. Proceedings. PowerCon 2000, 2000, vol. 1, pp. 55–59 vol.1.

[2] F. Giraud and Z. M. Salameh, "Steady-state performance of a grid-connected rooftop hybrid wind-photovoltaic power system with battery storage," IEEE Transactions on Energy Conversion, vol. 16, no. 1, pp. 1–7, 2001.

[3] T. Xing-guo, W. Hui, and L. Qing-min, "Multi-port topology for composite energy storage and its control strategy in micro-grid," in Power Electronics and Motion Control Conference (IPEMC), 2012 7th International, 2012, vol. 1, pp. 351–355.

[4] C. Wang and M. H. Nehrir, "Power Management of a Stand-Alone Wind/Photovoltaic/Fuel Cell Energy System," IEEE Transactions on Energy Conversion, vol. 23, no. 3, pp. 957–967, 2008.

[5] R. Majumder, A. Ghosh, G. Ledwich, and F. Zare, "Power Management and Power Flow Control With Back-to-Back Converters in a Utility Connected Microgrid," IEEE Transactions on Power Systems, vol. 25, no. 2, pp. 821–834, 2010.

[6] R. Majumder, "A Hybrid Microgrid With DC Connection at Back to Back Converters," IEEE Transactions on Smart Grid, vol. Early Access Online, 2013.

[7] J. Niiranen, R. Komsi, M. Routimo, T. Lähdeaho, and S. Antila, "Experiences from a back-to-back converter fed village microgrid," in Innovative Smart Grid Technologies Conference Europe (ISGT Europe), 2010 IEEE PES, 2010, pp. 1–5.

[8] B. Morvaj, L. Lugaric, and S. Krajcar, "Demonstrating smart buildings and smart grid features in a smart energy city," in Proceedings of the 2011 3rd International Youth Conference on Energetics (IYCE), 2011, pp. 1–8.

[9] M. Tasdighi, P. Jambor Salamati, A. Rahimikian, and H. Ghasemi, "Energy management in a smart residential building," in 2012 11th International Conference on Environment and Electrical Engineering (EEEIC), 2012, pp. 128–133.

[10] L. Wang, Z. Wang, and R. Yang, "Intelligent Multiagent Control System for Energy and Comfort Management in Smart and Sustainable Buildings," IEEE Transactions on Smart Grid, vol. 3, no. 2, pp. 605–617, 2012.

[11] J. Byun and S. Park, "Development of a self-adapting intelligent system for building energy saving and context-aware smart services," IEEE Transactions on Consumer Electronics, vol. 57, no. 1, pp. 90–98, 2011.

[12] G. T. Costanzo, G. Zhu, M. F. Anjos, and G. Savard, "A System Architecture for Autonomous Demand Side Load Management in Smart Buildings," IEEE Transactions on Smart Grid, vol. 3, no. 4, pp. 2157–2165, 2012.

[13] X. Guan, Z. Xu, and Q.-S. Jia, "Energy-Efficient Buildings Facilitated by Microgrid," IEEE Transactions on Smart Grid, vol. 1, no. 3, pp. 243–252, 2010.

[14] J. Bryan, R. Duke, and S. Round, "Decentralized generator scheduling in a nanogrid using DC bus signaling," in IEEE Power Engineering Society General Meeting, 2004, 2004, pp. 977–982 Vol.1.

[15] M. Pipattanasomporn, M. Kuzlu, and S. Rahman, "An Algorithm for Intelligent Home Energy Management and Demand Response Analysis," IEEE Transactions on Smart Grid, vol. 3, no. 4, pp. 2166–2173, 2012.

[16] J.-H. Jeon, J.-Y. Kim, H.-M. Kim, S.-K. Kim, C. Cho, J.-M. Kim, J.-B. Ahn, and K.-Y. Nam, "Development of Hardware In-the-Loop Simulation System for Testing Operation and Control Functions of Microgrid," IEEE Transactions on Power Electronics, vol. 25, no. 12, pp. 2919–2929, 2010.

[17] V. K. Sood, D. Fischer, J. M. Eklund, and T. Brown, "Developing a communication infrastructure for the Smart Grid," in 2009 IEEE Electrical Power Energy Conference (EPEC), 2009, pp. 1–7.

[18] K. O. Oureilidis and C. S. Demoulias, "Microgrid wireless energy management with energy storage system," in Universities Power Engineering Conference (UPEC), 2012 47th International, 2012, pp. 1–6.

[19] L. K. Siow, P. L. So, H. B. Gooi, F. L. Luo, C. J. Gajanayake, and Q. N. Vo, "Wi-Fi based server in microgrid energy management system," in TENCON 2009 - 2009 IEEE Region 10 Conference, 2009, pp. 1–5.

[20] Y. Zhu, F. Zhuo, and L. Xiong, "Communication platform for energy management system in a master-slave control structure microgrid," inPower Electronics and Motion Control Conference (IPEMC), 2012 7th International, 2012, vol. 1, pp. 141–145.

[21] N. Saito, "Ecological Home Network: An Overview," Proceedings of the IEEE, vol. Early Access Online, 2013.

Parte II ESTUDIO

ECONÓMICO

Capítulo 1 ESTUDIO ECONÓMICO

El mercado mundial de los sistemas y dispositivos de gestión energética para hogares y empresas, tiene unas perspectivas de crecimiento sostenido en la próxima década, con tasas medias cercanas al 6% anual, que según los expertos se pueden ver superadas año tras año. Esto se debe fundamentalmente a dos factores:

- Las energías renovables se vislumbran como una de las palancas, que permitirán superar la actual crisis económica e iniciar un nuevo ciclo alcista.

- La necesidad de conseguir un consumo energético racional y sostenible, ya que el modelo energético actual no es capaz de mantenerse indefinidamente y es preciso acceder de forma escalonada a fuentes de energía renovable, procurando mezclar las energías tradicionales procedentes del petróleo y fisión nuclear, con las nuevas energías renovables.

Si nos centramos en el mercado español estas perspectivas mundiales pueden tener gran validez, ya que nuestro país cuenta con una climatología favorable para la producción de energías renovables y actualmente se encuentra en una posición de liderazgo en este campo energético.

Un producto como el desarrollado en este proyecto fin de carrera, entronca perfectamente en este escenario. Esto se debe a que permite a un hogar o a una empresa seguir usando la energía eléctrica tradicional, al tiempo que se compatibiliza este uso con la incorporación a su consumo eléctrico de una energía renovable. Con el atractivo de que se reducirá el importe pagado a la compañía eléctrica que le suministre la energía. Se puede decir que el principio es el mismo que se emplea actualmente en los coches híbridos.

Dentro de la factura de la luz tradicional el ahorro de este producto incide en dos apartados: en el número de kilovatios hora consumidos y en la potencia contratada. Uno de los condicionantes del producto es que se pueda instalar un generador de energía renovable, tipo paneles solares o molinos de viento. El trabajo se plantea, en principio, pensando en paneles solares, por lo que los clientes potenciales se centran en aquellos que puedan tener acceso al tejado de sus instalaciones para poner paneles solares o a aquellos con un terreno próximo donde ubicar los paneles.

Los ahorros alcanzados se pueden estimar en un 20% en la cuota fija relacionada con la potencia contratada y en un 33% en el consumo de kilovatios hora.Por este motivo en una primera etapa se puede pensar en los siguientes clientes:

Sector residencial: Viviendas aislada, chalets adosados y también comunidades de vecinos que pueden usar este tipo de productos para reducir la factura del recibo de la luz comunitaria.

Sector empresarial: Negocios ubicados en parques tecnológicos con posibilidad de instalar paneles solares y hoteles.

Si analizamos las etapas típicas del ciclo de vida de un producto en el mercado: introducción, crecimiento, madurez y declive, observamos, que en la etapa de introducción y crecimiento se pueden tener pérdidas, por lo que es necesario tener muy claro el segmento de mercado al que nos dirigimos.

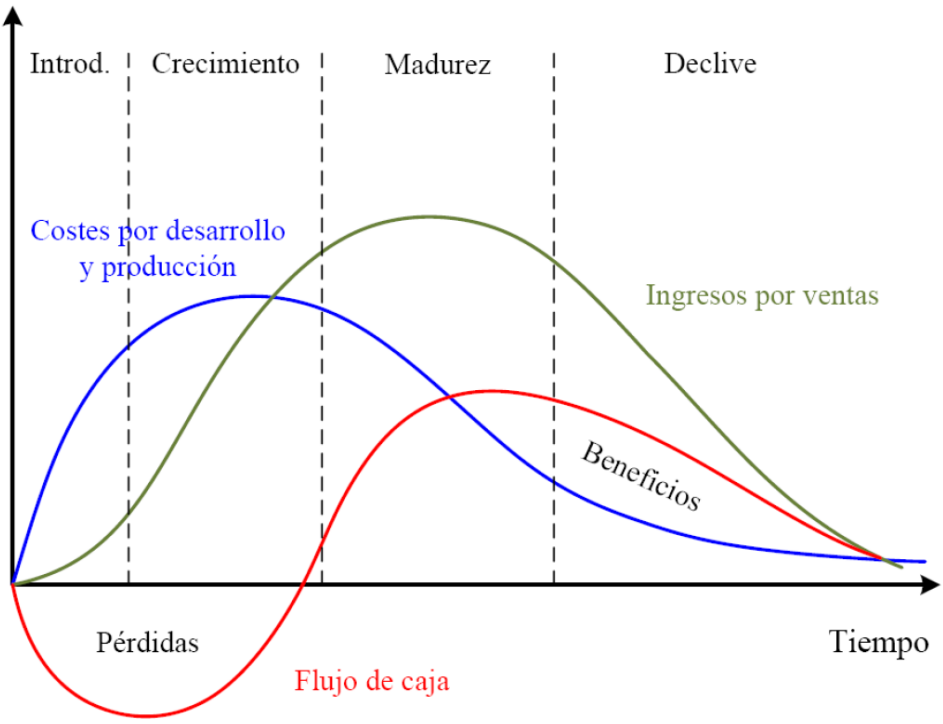

Figura 46. Ciclo de vida de un producto.

En el sector residencial hay que dirigirse a clientes tecnológicos de alto nivel adquisitivo, que es coherente con la condición de vivir en casas aisladas, que puedan asumir que la reducción en su factura del recibo de la luz no justifique plenamente la inversión inicial,

pero que sientan la satisfacción de pertenecer al club de los usuarios "híbridos" de energía eléctrica.

En el sector empresas hay que dirigirse a empresas relacionadas con el cuidado del medio ambiente y la sostenibilidad, ya que este tipo de clientes puede trasladar a su imagen corporativa este tipo de inversiones.

Como conclusión, el producto desarrollado en este proyecto fin de carrera se encuentra en un sector con tasas de crecimiento elevadas y con un mercado potencial en expansión, siendo preciso, en una primera etapa, seleccionar adecuadamente los nichos de mercado al que han de dirigirse los esfuerzos de venta.

Parte III MANUAL DE USUARIO

Capítulo 1 MANUAL DE USUARIO

En este capítulo se indica como encender el inversor, como apagarlo y qué hacer si se desea pasar del modo aislado al modo conectado a la red y viceversa.

Para encender correctamente el microinversor AN1444:

- Asegúrese de que el interruptor ON/OFF está en la posición de OFF.
- Conecte los cables a la entrada del microinversor, asegúrese que la polaridad es correcta (+/-).
- Conecte los cables al generador de corriente continua, manteniendo el generador apagado en todo momento.
- Conecte el cable de tensión alterna a las bornas de salida.
- Conecte el cable de tensión alterna a la red de distribución o a la microrred.
- Encienda el generador de corriente continua.
- Coloque el interruptor ON/OFF en la posición de ON.

Para apagar correctamente el microinversor AN1444:

- Coloque el interruptor ON/OFF en la posición de OF.
- Apague el generador de continua.
- Desconecte el cable de tensión alterna de la red de distribución o de la microrred.
- Desconecte el resto de cables.

Para pasar del modo de funcionamiento aislado desde el modo conectado a red no es necesario realizar ninguna acción. Cuando la tensión baje del nivel mínimo, se realizará de manera automática.

Para pasar del modo aislado al modo conectado a red:

- Apague el microinversor.
- Conecte su microrred a la red de distribución.
- Encienda el microinversor.

Si en cualquiera de los procesos anteriores o durante el funcionamiento del microinversor observa que parpadea el LED rojo D27, ha ocurrido un error. Apunte el número de parpadeos y apague el microinversor. A continuación se muestra una tabla con la equivalencia entre el número de parpadeos el error que ha ocurrido, véase la Tabla 5.

Número de parpadeos	Error
1	Tensión de entrada fuera de los límites de funcionamiento.
2	Frecuencia de la red inapropiada.
3	Tensión de la red fuera de los límites de funcionamiento.
4	Sobre corriente en el inversor Full Bridge.
5	Sobre corriente en el convertidor Flyback DC/DC.
6	Temperatura elevada.
7	Tensión de alimentación interna inapropiada.
8	Sobre tensión a la salida del convertidor Flyback DC/DC.
9	Tensión de referencia 2.5 V inapropiada.
10	Offset de la corriente alterna excesivo.
11	No se produce paso por cero en la tensión de la red.

Tabla 5. Errores posibles del microinversor.

Parte IV CÓDIGO FUENTE

Capítulo 1 DEFINES.H

En esta sección se muestra el código del archivo defines.h, en él se hacen todas las definiciones de constantes del programa.

```
#define ON 1
#define OFF 0

#define OPTO_DRV1 LATCbits.LATC0
#define OPTO_DRV2 LATCbits.LATC13

#define LED_DRV1 LATCbits.LATC12
#define LED_DRV2 LATCbits.LATC3
#define LED_DRV3 LATCbits.LATC8

#define FLYBACKPERIOD 16864
//~56kHz switching frequency(1/56kHz/1.06ns)= 16864
#define FULLBRIDGEPERIOD (FLYBACKPERIOD >> 2)
//228kHz switching frequency (1/228kHz/1.04ns) = 4216
#define FLYBACKINTERLEAVEDPHASE (FLYBACKPERIOD >> 1)
// 180Deg Phase-shift (flybackPeriod/2)

#define FLYBACKALTDTR 150 // Dead-time
#define FLYBACKDTR 50

#define FLYBACKDUTY1 PDC1 // Re-define PWM Duty Cycle Registers
#define FLYBACKDUTY2 PDC2

#define MAXDUTYCLAMPED 24575
// Maximum duty cycle of 75% in Q15 format

// 180 degree is equal to 32767
#define ONEHUNDREDSEVENTYFIVEDEGREE 31800
// Degree to turn off the Full-Bridge unfolding circuit
#define NINETYDEGREE 16383
// Vac Operating Voltage Limits
#define INVERTER_OVERVOLTAGE_LIMIT 14800
// Inverter Output Voltage ((264 * sqrt(2)) / turns ratio of TR1)*(gain of u6)
// Converted into ADC counts and Q15 format minus the offset
#define INVERTER_OVERVOLTAGE_LIMIT_HYS 14400
// Inverter Output Over Voltage Hysteresis ~3Vac RMS

#define INVERTER_UNDERVOLTAGE_LIMIT 11400
// Inverter Output Voltage ((210 * sqrt(2)) / turns ratio of TR1)*(gain of u6)
// Converted into ADC counts and Q15 format minus the offset
#define INVERTER_UNDERVOLTAGE_LIMIT_HYS 11700
// Inverter Output Under Voltage Hysteresis ~3Vac RMS

// PV Panel Operating Voltage Limits
#define PVPANEL_MPP_LIMIT 26900
// Max MPP Voltage ~46Vdc
```

```
#define PVPANEL_OVERVOLTAGE_LIMIT 30370
// ~52V open circuit voltage, 52 * (R74/(R74+R72))
// Converted into ADC counts and Q15 format
#define PVPANEL_OVERVOLTAGE_LIMIT_HYS 28900
// PV Panel Over Voltage Hysteresis ~2V

#define PVPANEL_UNDERVOLTAGE_LIMIT 10900
// Minimum Operating Voltage ~20Vdc, 18.5 * (R74/(R74+R72))
// Converted into ADC counts and Q15 format
#define PVPANEL_UNDERVOLTAGE_LIMIT_HYS 12300
// PV Panel Under Voltage Hysteresis ~2V

// Inverter Output Current Max Rating
#define INVERTER_OUTPUTCURRENT_MAX 13000
// At 210Vac full load, expected peak inverter output current ~1.6A.
// This current * gain of 185mV/A * gain of U5, measured by the ADC
// and converted to Q15. Subtract off the offset
// AC Current Offset Limits
#define MINOFFSETCURRENT 12896
// Nominal ~1.6V, convert to ADC and Q15
#define MAXOFFSETCURRENT 17856

// Maximum Output Power Limit
#define DERATINGFACTOR 39600
#define DERATINGSLOPE 25000//25673 // 1.567 in Q14 format
#define DERATINGCONSTANT 18235 // 1.113 in Q14 format

// Power De-rating Limits
#define POWERDERATINGLIMIT 14300 // Corresponds to 24.5Vdc

// Power Decrement Factor for anti-islanding
#define POWERDECREMENTFACTOR Q15(.02)
// This is ~2% which is an initial power de-rating

// Maximum Average Flyback Current Limit
#define MAXFLYBACKCURRENT 29000
// Max average current through flyback stage ~3V on ADC

// 12V Drive Supply Operating Limits
#define MAXDRIVEVOLTAGE 30766
// 12.5V drive supply * (R2/(R2+R1)), convert to ADC and then Q15
#define MINDRIVEVOLTAGE 28300 // 11.5V in Q15 format

// Temperature Operating Limits
#define MAXTEMPERATURE 12400
// Vout = (75C * 10mV/C) + .5V = 1.25V, Convert to ADC reading then Q15
#define MAXTEMPERATURE_HYS 11400
// 65C degrees, Convert to ADC reading and then Q15

// 2.5V Operating Limits
#define MINREFERENCEVOLTAGE 15744
// 2.4V * (R141/(R141+R128)), convert to ADC and then convert to Q15
#define MAXREFERENCEVOLTAGE 17056
// 2.6V in Q15 format

// 50/60Hz Operating Limits
#define INVERTERPERIOD50HZMIN 528
// Frequency set points for 50Hz (47Hz - 53Hz), ADC ISR rate is 17.87us
#define INVERTERPERIOD50HZMAX 595
```

```
#define INVERTERPERIOD60HZMIN 461
// Frequency set points for 60Hz (59.3Hz - 60.5Hz), ADC ISR rate is 17.87us
#define INVERTERPERIOD60HZMAX 473
#define INVERTERPERIODHYS 3 // Hysteresis for Inverter frequency


// MPPT Definitions
#define MPPTFACTORMINIMUM 700 // Minimum MPPT Factor to start the inverter
#define MININCREMENTMPPTFACTOR 20
// Values should be small enough to avoid large voltage
#define MAXINCREMENTMPPTFACTOR 30 // deviations when operating at MPP
#define MINDECREMENTMPPTFACTOR 20
#define MAXDECREMENTMPPTFACTOR 30
#define LARGEVOLTAGEDIFFERENCE 8

// Burst Mode Definitions
#define BURSTMODECOUNT 6700
// Burst Mode Count (~1 minute) - 1/(2xVac) * counter
#define BURSTMODETHRESHOLDLOW 1600
// Power to which burst mode is applied < ~15%
#define BURSTMODETHRESHOLDHIGH 2800
// Power to which burst mode is removed ~18%

// Various #defines
#define ZEROCROSSCOUNT 60
// Number of zero cross events required before switching to Day Mode
#define RESTARTCOUNT 10000
// 10000 * 100us - 1s of no faults to switch to system startup
#define CRITICALFAULTCOUNT 20000
// 20000 * 100us - 2s before trying to restart the system
#define LOADBALCOUNT 2
// Rate to execute the load balance routine in the state machine
#define PVPANEL_40V 23341
// Panel Voltage 40V, used for changing CMP current reference
#define PVPANEL_30V 17500
// Panel Voltage 30V, used for changing CMP current reference
#define PVPANEL_VOLTAGEDROP -600
// Change in PV panel average voltage due to large AC Voltage Change
// PV Voltage change of ~0.5V * (7.5/127.5), convert to ADC and Q15

// Coefficients for PI Controller
#define Ra Q15(0.18)
#define Rsa Q15(0.02)

// Coefficients for Load Balancing
#define MAXBALANCE Q15(0.01)
#define KAQ15 Q15(0.065)
#define KSAQ15 Q15(0.01)

// System State Definitions
#define SYSTEMSTARTUP 0
#define DAYMODE 1
#define SYSTEMERROR 2
#define ISLANDSTARTUP 3
#define ISLANDMODE 4

// Input Switch State
#define SWITCHOFF 0
#define SWITCHON 1
```

```
// Fault State Definitions
#define NO_FAULT 0
#define PV_PANEL_VOLTAGE 1
#define INVERTER_FREQUENCY 2
#define INVERTER_VOLTAGE 3
#define INVERTER_OVERCURRENT 4
#define FLYBACK_OVERCURRENT 5
#define TEMPERATURE 6
#define DRIVE_SUPPLY 7
#define FLYBACK_OUTPUT_VOLTAGE 8
#define REFERENCE_VOLTAGE 9
#define ACCURRENT_OFFSET 10
#define HARDWAREZEROCROSS 11

// State for Full Bridge Drive
#define FULLBRIDGE_Q3Q4_ACTIVE 1
#define FULLBRIDGE_INACTIVE_2ND_QUADRANT 2
#define FULLBRIDGE_Q2Q5_ACTIVE 3
#define FULLBRIDGE_INACTIVE_4TH_QUADRANT 4

//#define BENCHTESTING
#ifdef BENCHTESTING
#define MPPTFACTOR_BENCHTESTING 9900
#define MPPTFACTORINCREMENT 20
// Softstart increments MPPT by MPPTFACTORINCREMENT every 10ms
#define MPPTCOUNT 100 // Slow down the increment rate to -> 100 * 100us
#endif
```

Capítulo 2 INIT.C

En esta sección se muestra el código del archivo init.c, en él se hacen todas las inicializaciones de funciones del programa.

```
void initClock(void)
{
// Configure Oscillator to operate the device at 40 MIPS
// Fosc = Fin*M/(N1*N2), Fcy = Fosc/2
// Fosc = 7.37*(43)/(2*2) = 80MHz for Fosc, Fcy = 40MHz

// Configure PLL prescaler, PLL postscaler, PLL divisor
PLLFBD = 41; // M = PLLFBD + 2
CLKDIVbits.PLLPOST = 0; // N1 = 2
CLKDIVbits.PLLPRE = 0; // N2 = 2

// Change oscillator to FRC + PLL
__builtin_write_OSCCONH(0x01); // New Oscillator FRC w/ PLL
__builtin_write_OSCCONL(0x01); // Enable Clock Switch
while(OSCCONbits.COSC != 0b001); // Wait for new Oscillator to become FRC w/ PLL
while(OSCCONbits.LOCK != 1); // Wait for Pll to Lock

// Now setup the ADC and PWM clock for 120MHz
// ((FRC * 16) / APSTSCLR ) = (7.37 * 16) / 1 = ~ 120MHz

ACLKCONbits.FRCSEL = 1; // FRC provides input for Auxiliary PLL (x16)
ACLKCONbits.SELACLK = 1;
// Auxiliary Ocillator provides clock source for PWM & ADC
ACLKCONbits.APSTSCLR = 7; // Divide Auxiliary clock by 1
ACLKCONbits.ENAPLL = 1; // Enable Auxiliary PLL

while(ACLKCONbits.APLLCK != 1); // Wait for Aux. PLL to Lock
}


void initPWM(void)
{
// PWM1 and PWM2 Configuration for Interleaved Flyback Drive
PTCON2bits.PCLKDIV = 0; // Maximum (1.06nsec) timing resolution

PTPER = FLYBACKPERIOD; // Flyback inverter period value
PHASE2 = FLYBACKINTERLEAVEDPHASE; // 180Deg phase-shift for second converter

ALTDTR1 = FLYBACKALTDTR; // Setup Deadtime for PWM1 and PWM2
DTR1 = FLYBACKDTR;
ALTDTR2 = FLYBACKALTDTR;
DTR2 = FLYBACKDTR;
PDC1 = 0; // Initialize Flyback duty cycle
PDC2 = 0;
PWMCON1bits.ITB = 0;
// PTPER register provides the timing for PWM1, PWM2 generator
PWMCON2bits.ITB = 0;

PWMCON1bits.IUE = 1; // Enable immediate updates
```

```
PWMCON2bits.IUE = 1;

PWMCON1bits.DTC = 0; // Positive dead-time enabled
PWMCON1bits.DTC = 0;
IOCON1bits.PMOD = 0; // PWM1H and PWM1L is in complementary output mode
IOCON2bits.PMOD = 0; // PWM2H and PWM2L is in complementary output mode

IOCON1bits.POLL = 1; // PWM1L and PWM2L have inverted polarity (active-low)
IOCON2bits.POLL = 1; // to drive the P-channel MOSFETs

IOCON1bits.PENH = 1; // PWM controls I/O pins
IOCON1bits.PENL = 1;
IOCON2bits.PENH = 1;
IOCON2bits.PENL = 1;

IOCON1bits.OVRDAT = 0; // Flyback override data (Safe-State)
IOCON1bits.OVRENH = 1; // Enable override (Disable MOSFETs)
IOCON1bits.OVRENL = 1;
IOCON2bits.OVRDAT = 0; // Flyback override data (Safe-State)
IOCON2bits.OVRENH = 1; // Enable override (Disable MOSFETs)
IOCON2bits.OVRENL = 1;

// Leading Edge Blanking for Current Protection

// PWM1H rising and falling edges trigger LEB counter (240ns)
// LEB is applied to current limit input
LEBCON1 = 0xC4F0;

// PWM2H rising and falling edges trigger LEB counter (240ns)
// LEB is applied to current limit input
LEBCON2 = 0xC4F0;

// PWM Fault/Current limit Setup
FCLCON1bits.FLTMOD = 0; // Latched Fault Mode
FCLCON1bits.CLMOD = 1; // Current Limit Enabled
FCLCON1bits.CLSRC = 1; // Current Limit Source is Fault 2
FCLCON1bits.CLPOL = 0; // Current Limit Source is Active High

FCLCON2bits.FLTMOD = 0; // Latched Fault Mode
FCLCON2bits.CLMOD = 1; // Current Limit Enabled
FCLCON2bits.CLSRC = 2; // Current Limit Source is Fault 3
FCLCON2bits.CLPOL = 0; // Current Limit Source is Active High

// Triggers for ADC Module
TRGCON1bits.TRGDIV = 0; // Trigger ADC every PWM Period
TRGCON1bits.TRGSTRT = 0; // Wait 0 PWM cycles before generating first PWM
trigger

TRGCON2bits.TRGDIV = 0; // Trigger ADC every PWM Period
TRGCON2bits.TRGSTRT = 0; // Wait 0 PWM cycles before generating first PWM
trigger

TRIG1 = 8; // PWM1 used to trigger ADCP4
TRIG2 = 8;


// PWM3 Configuration for Full-Bridge Drive
PWMCON3bits.ITB = 1; // Independent Time Base

PHASE3 = FULLBRIDGEPERIOD; // 2x the switching frequency of the flyback
SPHASE3 = FULLBRIDGEPERIOD;
```

```
PDC3 = PHASE3 >> 1; // 50% duty cycle
SDC3 = SPHASE3 >> 1;

PWMCON3bits.DTC = 3; // Dead-time disabled

IOCON3bits.PMOD = 1; // Redundant Output Mode

IOCON3bits.PENH = 1; // PWM controls I/O pins
IOCON3bits.PENL = 1;

IOCON3bits.OVRDAT = 0; // Full-Bridge override data (Safe State)
IOCON3bits.OVRENH = 1; // Enable override (Disable MOSFETs)
IOCON3bits.OVRENL = 1;

// PWM3 Fault/Current Limit Setup
FCLCON3bits.FLTMOD = 3; // Disable Faults

}


void initADC(void)
{
ADCONbits.SLOWCLK = 1; // Requirement from ADC Errata
ADCONbits.FORM = 0; // Output in Integer Format
ADCONbits.EIE = 0; // Disable Early Interrupt
ADCONbits.ORDER = 0; // Even channel first
ADCONbits.SEQSAMP = 0; // Simultaneus Sampling Enabled
ADCONbits.ASYNCSAMP = 1; // Asynchronous Sampling Enabled
ADCONbits.ADCS = 4; // Clock Divider is set up for Fadc/5, TAD = 41.6ns

// For simultaneus sampling total conversion time for one pair is 0.58us

ADSTAT = 0; // Clear the ADSTAT register
ADPCFGbits.PCFG0 = 0; // AN0 PV Panel Voltage Sense
ADPCFGbits.PCFG1 = 0; // AN1 PV current sense of Interleaved Flyback 1 converter
ADPCFGbits.PCFG2 = 0; // AN2 PV current sense of Interleaved Flyback 2 converter
ADPCFGbits.PCFG3 = 0; // AN3 Output AC Current Sense
ADPCFGbits.PCFG6 = 0; // AN6 2.5V Reference
ADPCFGbits.PCFG7 = 0; // AN7 Output AC Voltage Sense
ADPCFGbits.PCFG10 = 0; // AN10 Temperature Sense
ADPCFGbits.PCFG11 = 0; // AN11 12V Drive Supply Sense
ADCPC0bits.TRGSRC0 = 4; // AN0 and AN1 triggered by PWM1
ADCPC0bits.TRGSRC1 = 5; // AN2 and AN3 triggered by PWM2
ADCPC1bits.TRGSRC3 = 5;
// AN6 and AN7 triggered by PWM2 (use this ISR for Control Loop)

ADCPC2bits.TRGSRC5 = 5; // AN10 and AN11 triggered by PWM2

IPC28bits.ADCP3IP = 6;
// Set ADC Pair3 Interrupt Priority (2nd highest priority)

IFS7bits.ADCP3IF = 0; // Clear ADC Pair3 Interrupt Flag
IEC7bits.ADCP3IE = 1; // Enable ADC Pair3 Interrupt at start
}



void initIOPorts(void)
{
// PWM1 and PWM2 Fault/CMP Remap to Virtual I/O Pins
RPINR30bits.FLT2R = 32;
```

```
RPINR30bits.FLT3R = 33;

RPOR16bits.RP32R = 0b101000;
RPOR16bits.RP33R = 0b101001;

//LED_DRV1
TRISCbits.TRISC12 = 0;
LED_DRV1 = OFF;

//LED_DRV2
TRISCbits.TRISC3 = 0;
LED_DRV2 = OFF;

//LED_DRV3
TRISCbits.TRISC8 = 0;
LED_DRV3 = OFF;

//OPTO_DRV1
TRISCbits.TRISC0 = 0;
OPTO_DRV1 = 0;

//OPTO_DRV2
TRISCbits.TRISC13 = 0;
OPTO_DRV2 = 0;

//GPIO State when PWM doesn't have ownership
LATAbits.LATA3 = 0;
LATAbits.LATA4 = 0;
LATBbits.LATB11 = 0;
LATBbits.LATB12 = 0;
LATBbits.LATB13 = 0;
LATBbits.LATB14 = 0;
}

//Timer to get time in order to create model signal for isolated mode
void initIsolatedTimer(void)
{
T1CONbits.TCKPS = 0; // Prescaler of 1:1
PR1 = 2040;          // (51us /25ns) = 2040
IPC0bits.T1IP = 5; // Set up Timer interrupt priority
IFS0bits.T1IF = 0; // Clear Interrupt Flag
IEC0bits.T1IE = 1; // Enable Timer2 interrupt
}

void initStateMachineTimer(void)
{
T2CONbits.TCKPS = 0; // Prescaler of 1:1
PR2 = 4000; // (100us /25ns) = 4000
IPC1bits.T2IP = 4; // Set up Timer interrupt priority
IFS0bits.T2IF = 0; // Clear Interrupt Flag
IEC0bits.T2IE = 1; // Enable Timer2 interrupt
}

void initLedFaultIndicatorTimer(void)
{
T3CONbits.TCKPS = 3; // Prescaler of 1:256
PR3 = 46875; // (300ms /(25ns*256)) = 46875
IPC2bits.T3IP = 5; // Set up Timer interrupt priority
IFS0bits.T3IF = 0; // Clear Interrupt Flag
IEC0bits.T3IE = 1; // Enable Timer2 interrupt when fault is present
}
```

```
void initCMP(void)
{
//CMP2C (Flyback Current Phase 1)
CMPCON2bits.EXTREF = 0; // Internal Voltage Selected
CMPCON2bits.RANGE = 1; // Select Avdd/2 (high-range)
CMPCON2bits.INSEL = 2; // Select Comparator 2C

CMPDAC2bits.CMREF = 700; // CMREF * (Avdd/2) / 1024 - Volts
CMPCON2bits.CMPON = 1; // Enable CMP2

IPC25bits.AC2IP = 7; // Highest Priority Interrupt
IFS6bits.AC2IF = 0; // Clear Interrupt Flag
IEC6bits.AC2IE = 1; // Enable Interrupt to declare fault

//CMP3B (Flyback Current Phase 2)
CMPCON3bits.EXTREF = 0; // Internal Voltage Selected
CMPCON3bits.RANGE = 1; // Select Avdd/2 (high-range)
CMPCON3bits.INSEL = 1; // Select Comparator 3B

CMPDAC3bits.CMREF = 700; // CMREF * (Avdd/2) / 1024 - Volts
CMPCON3bits.CMPON = 1; // Enable CMP3

IPC26bits.AC3IP = 7; // Highest Priority Interrupt
IFS6bits.AC3IF = 0; // Clear Interrupt Flag
IEC6bits.AC3IE = 1; // Enable Interrupt to declare fault


//CMP4C (Flyback Over Voltage Protection)
CMPCON4bits.EXTREF = 0; // Internal Voltage Selected
CMPCON4bits.RANGE = 1; // Select Avdd/2 (high-range)
CMPCON4bits.INSEL = 2; // Select Comparator 4C

CMPDAC4bits.CMREF = 1000; // CMREF * (Avdd/2) / 1024 - 1.6 Volts
CMPCON4bits.CMPON = 1; // Enable CMP4

IPC26bits.AC4IP = 7; // Highest Priority Interrupt
IFS6bits.AC4IF = 0; // Clear Interrupt Flag
IEC6bits.AC4IE = 1; // Enable Interrupt to disable PWMs and set fault

}
```

Capítulo 3 MAIN.C

En esta sección se muestra el código del archivo main.c, en él se recoge el programa principal del inversor.

```
#include "Solar Microinverter_main.h"

// Configuration bits
// Start-up with FRC and switch to FRC w/ Pll
// Configure Watch Dog Timeout (Software Enable/Disable)
// Select ICSP Pair 2 for debugging/programming
_FOSCSEL(FNOSC_FRC)
_FOSC(FCKSM_CSECMD & OSCIOFNC_ON & IOL1WAY_OFF)
_FWDT(FWDTEN_OFF & WDTPOST_PS16 & WDTPRE_PR32 & WINDIS_OFF)
_FPOR(FPWRT_PWR128)
_FICD(ICS_PGD2 & JTAGEN_OFF)

int main()
{
initClock();
// Initialize Device Oscillator and Auxiliary Oscillator (PWM/ADC)

initIOPorts(); // Initialize all I/O Ports

initADC(); // Initialize ADC Module:
ADCONbits.ADON = 1;
// Enable the ADC Module early for ADC Settling Time

initCMP(); // Initialize Comparator Module
initPWM(); // Initialize PWM Module
initIsolatedTimer(); // Initialize Timer1 (Isolated Mode)
initStateMachineTimer(); // Initialize Timer2 (State Machine)
initLedFaultIndicatorTimer(); // Initialize Timer3 (Fault Indication)
T1CONbits.TON = 1; // Enable Timer 1
T2CONbits.TON = 1; // Enable State Machine Timer
PTCONbits.PTEN = 1; // Enable PWM Module

RCONbits.SWDTEN = 1; // Enable WDT (cleared at every zero cross)
while(1)
{
Nop();
Nop();
Nop();
}
return 0;
}
```

Capítulo 4 ISR.C

En esta sección se muestra el código del archivo isr.c, en él se reciben datos del conversor A/D y se realizan varias operaciones con ellos.

```
#include "Solar Microinverter_isr.h"
// System Faults
unsigned char criticalFaultFlag = 0, inverterFrequencyErrorFlag = 0, frequencyFaultCounter
= 0;
int maxInverterOutputVoltage = 0, rectifiedInverterOutputVoltage = 0;


// Compensator Variables
int rectifiedInverterOutputCurrent = 0, peakInverterOutputVoltage = 0;
long int Ioutput = 0;
unsigned char fullBridgeState = FULLBRIDGE_INACTIVE_4TH_QUADRANT;


// Inverter Current Reference Variables
unsigned char ninetyDegreeDetectFlag = 0;
unsigned int globalAngle = 0, sineAngle = 0, deltaAngle = 0;
unsigned int currentReferenceDynamic = 0;


// ADC Variables
unsigned int pvPanelVoltage = 0, flybackCurrent1 = 0, flybackCurrent2 = 0;
int inverterOutputVoltage = 0, inverterOutputCurrent = 0, prevInverterOutputVoltage = 0;
unsigned int driveSupplyVoltage = 0, measuredTemperature = 0, referenceVoltage = 0;


// Flyback Current Moving Average Variables/Arrays
unsigned char currentArrayCnt = 0;
unsigned int averageFlybackCurrent1 = 0, averageFlybackCurrent2 = 0;
unsigned int flybackCurrent1Array[8] = {0,0,0,0,0,0,0,0};
unsigned int flybackCurrent2Array[8] = {0,0,0,0,0,0,0,0};
long unsigned int flybackCurrent1Sum = 0, flybackCurrent2Sum = 0;


// AC Current Variables and Moving Avg Variables/Array
unsigned int averageRectifiedCurrent = 0, maxInverterOutputCurrent = 0;
unsigned int rectifiedInverterOutputCurrentArray[8] = {0,0,0,0,0,0,0,0};
long unsigned int rectifiedInverterOutputCurrentSum = 0;


// Peak Power and Burst Mode Variables
unsigned int averagePeakOutputPower = 0, peakInverterOutputCurrent = 0, peakOutputPower =
0;
unsigned char peakOutputPowerArrayCnt = 0;
unsigned int peakOutputPowerArray[8] = {0,0,0,0,0,0,0,0};
long unsigned int peakOutputPowerSum = 0;
unsigned char burstModeActiveFlag = 0;
unsigned int burstModeActiveCounter = 0;


// Zero Cross Variables
unsigned char zeroCrossDetectFlag = 0, zeroCrossCount = 0, firstQuadrantFlag = 0;
unsigned char thirdQuadrantFlag = 0, avgInputDataReadyFlag = 0;
unsigned char zcCounter = 0, startupZeroCrossCounter = 0;
unsigned char hardwareZeroCrossCounter = 0, prevFullBridgeState = 0;
unsigned int inverterPeriod = 0, inverterPeriodCounter = 0, prevInverterPeriod = 0,
numberofSamples = 0;


// MPPT Variables
unsigned int mpptFactorMaximum = 0, mpptDeratingFactor = 0;
```

```
long unsigned int inputCurrentSum = 0, inputVoltageSum = 0;

// Externally Defined Variables
extern unsigned char switchState, systemState, faultState, zeroCrossDelay;
extern      unsigned      char      inverterFrequencyState,      criticalFaultRestartFlag,
startFullBridgeFlag;
extern unsigned char zeroCrossDelayNom, zeroCrossDelayMin, zeroCrossDelayMax;
extern unsigned int mpptFactor, inputVoltageAverage, inputCurrentAverage;
extern unsigned int inverterPeriodMin, inverterPeriodMax, inverterOutputOverCurrent;
extern int deltaDutyCycle, acCurrentOffset;



// Variable Declaration for DMCI Debugging Tool
#ifdef DMCI_ISR
int array1[100];
int array2[100];
int array3[100];
unsigned char dmciArrayIndex = 0;
unsigned int dmciCount = 0;
#endif

// Variable Declaration for Isolated Mode
long unsigned int t =0;
double wred = 314.1593;
double amplitud = 325.2691;
double tred = 0.0000;
double angulo = 0.000;
double voltaje = 0.0000;
int ADCBUF7Isolated = 0;
extern long unsigned int ticks;
int ADCBUF7max=0;
int ADCBUF7min=1000;
int inverterOutputVoltageFalse = 0, prevInverterOutputVoltageFalse = 0;
int rectifiedInverterOutputVoltageFalse = 0, peakInverterOutputVoltageFalse = 0;

char paso=0;

void __attribute__((interrupt, no_auto_psv)) _CMP2Interrupt()
{
if((criticalFaultFlag == 1) && (criticalFaultRestartFlag == 1))
{
PTCONbits.PTEN = 0;
RCONbits.SWDTEN = 0; // Need to disable WDT
}

// As PWM is latched we need to set the faultState to Flyback overcurrent no matter what
// so that the latched fault can be removed
criticalFaultFlag = 1;
faultState = FLYBACK_OVERCURRENT;

IFS6bits.AC2IF = 0;
}

void __attribute__((interrupt, no_auto_psv)) _CMP3Interrupt()
{
if((criticalFaultFlag == 1) && (criticalFaultRestartFlag == 1))
{
PTCONbits.PTEN = 0;
RCONbits.SWDTEN = 0; // Need to disable WDT
}

// As PWM is latched we need to set the faultState to Flyback overcurrent no matter what
// so that the latched fault can be removed
criticalFaultFlag = 1;
faultState = FLYBACK_OVERCURRENT;
```

```
IFS6bits.AC3IF = 0;
}

void __attribute__((interrupt, no_auto_psv)) _CMP4Interrupt()
{
// Flyback Output Over Voltage Condition used only for protection against
// Full-Bridge failure (in normal operating conditions this event should never occur)

// Over-ride the flyback MOSFETs immediately
IOCON1bits.OVRENH = 1;
IOCON1bits.OVRENL = 1;
IOCON2bits.OVRENH = 1;
IOCON2bits.OVRENL = 1;

if((criticalFaultFlag == 1) && (criticalFaultRestartFlag == 1))
{
PTCONbits.PTEN = 0;
RCONbits.SWDTEN = 0; // Need to disable WDT
}

if(faultState == NO_FAULT)
{
criticalFaultFlag = 1;
faultState = FLYBACK_OUTPUT_VOLTAGE;
}

IFS6bits.AC4IF = 0;
}



void __attribute__((interrupt, no_auto_psv)) _ADCP3Interrupt()
{
// LED_DRV3 = ON;
// Read Flyback Current, PV Voltage, and 2.5V Reference in Q15 Format
pvPanelVoltage = (ADCBUF0 << 5); // Read PV Panel Voltage
flybackCurrent1 = (ADCBUF1 << 5); // Read PV cell Current at Flyback leg1
flybackCurrent2 = (ADCBUF2 << 5); // Read PV cell Current at Flyback leg2
referenceVoltage = (ADCBUF6 << 5);
// Read 2.5V Reference Voltage (used for AC offset)

// Read inverter output voltage and inverter output current (Q14)
inverterOutputCurrent = (ADCBUF3 << 5) - acCurrentOffset;
//Si se encuentra en modo isla en vez de medir toma los valores de una tabla
if((systemState != ISLANDSTARTUP)&& (systemState != ISLANDMODE))
{
inverterOutputVoltage = (ADCBUF7 << 5) - referenceVoltage;
}
else
{
//Se calcula el tiempo
if(IEC0bits.T1IE == 1)
{
IEC0bits.T1IE = 0; // Disable Timer1 interrupt
t=ticks;
IEC0bits.T1IE = 1; // Enable Timer1 interrupt
}
// Se calcula el valor que debería tener la tensión/////////////////
tred=0.051*t;
angulo = wred*tred;
angulo = (angulo*0.08133); //511/(2*3.1416) = 81.33
angulo = ((int)angulo)%512;
voltaje = amplitud*((double)seno512[(int)angulo])/10000;
//Despeje de la ecuación
voltaje = (voltaje*5)/190;
voltaje = (voltaje*0.23)+2.5; //6.2/27=0.23
voltaje = (0.66*voltaje); //3.3/(1.69+3.3)=0.66
```

```
//Pasar a digital el valor
ADCBUF7Isolated = (int)(voltaje*310); //1023/3.3 = 310
ADCBUF7Isolated = ADCBUF7Isolated - 5;
//Simulación de la lectura
inverterOutputVoltageFalse = (ADCBUF7Isolated << 5) - referenceVoltage;
//Lectura de la realidad para control en lazo cerrado
inverterOutputVoltage = (ADCBUF7 << 5) - referenceVoltage;
}
// Rectified Inverter Output Voltage
if(inverterOutputVoltage >= 0)
{
rectifiedInverterOutputVoltage = inverterOutputVoltage;
}
else
{
rectifiedInverterOutputVoltage = (-inverterOutputVoltage);
}

// Find Peak Inverter Output Voltage
if(rectifiedInverterOutputVoltage > maxInverterOutputVoltage)
{
maxInverterOutputVoltage = rectifiedInverterOutputVoltage;
}

// Rectify AC current and check for over current condition on the ouput
if(inverterOutputCurrent >= 0)
{
rectifiedInverterOutputCurrent = inverterOutputCurrent;
}
else
{
rectifiedInverterOutputCurrent = (-inverterOutputCurrent);
}

// Find Peak Inverter Output Voltage False
if(inverterOutputVoltageFalse >= 0)
{
rectifiedInverterOutputVoltageFalse = inverterOutputVoltageFalse;
}
else
{
rectifiedInverterOutputVoltageFalse = (-inverterOutputVoltageFalse);
}
if(rectifiedInverterOutputVoltageFalse > peakInverterOutputVoltageFalse)
{
peakInverterOutputVoltageFalse = rectifiedInverterOutputVoltageFalse;
}




// Find Peak Inverter Output Current
if(rectifiedInverterOutputCurrent > maxInverterOutputCurrent)
{
maxInverterOutputCurrent = rectifiedInverterOutputCurrent;
}



// Moving Average of Flyback Currents for load sharing
flybackCurrent1Sum        =        flybackCurrent1Sum        +        flybackCurrent1        -
flybackCurrent1Array[currentArrayCnt];
averageFlybackCurrent1 = flybackCurrent1Sum >> 3;

flybackCurrent2Sum        =        flybackCurrent2Sum        +        flybackCurrent2        -
flybackCurrent2Array[currentArrayCnt];
averageFlybackCurrent2 = flybackCurrent2Sum >> 3;
flybackCurrent1Array[currentArrayCnt] = flybackCurrent1;
```

```
flybackCurrent2Array[currentArrayCnt] = flybackCurrent2;

// Moving Average of AC Current for AC Current Fault
rectifiedInverterOutputCurrentSum=rectifiedInverterOutputCurrentSum+rectifiedInverterOutputCurre
nt- rectifiedInverterOutputCurrentArray[currentArrayCnt];

averageRectifiedCurrent = rectifiedInverterOutputCurrentSum >> 3;

rectifiedInverterOutputCurrentArray[currentArrayCnt++] = rectifiedInverterOutputCurrent;

if(currentArrayCnt >= 8)
{
currentArrayCnt = 0;
}

// Check for Over Current Condition
if((averageFlybackCurrent1       >       MAXFLYBACKCURRENT)||(averageFlybackCurrent2       >
MAXFLYBACKCURRENT))
{
if(faultState == NO_FAULT)
{
faultState = FLYBACK_OVERCURRENT;
}
}

// Inverter Over Current Fault Check
// inverterOutputOverCurrent takes into account the delta in the acCurrentOffset
if((averageRectifiedCurrent>inverterOutputOverCurrent)&&((systemState==DAYMODE)||(systemState==I
SLANDMODE)))
{
if((criticalFaultFlag == 1) && (criticalFaultRestartFlag == 1))
{
PTCONbits.PTEN = 0;
RCONbits.SWDTEN = 0; // Need to disable WDT
}

if(faultState == NO_FAULT)
{
criticalFaultFlag = 1;
faultState = INVERTER_OVERCURRENT;
}
}

// Call zero cross detection function to look for both +ve and -ve zero cross events
// Routine also keeps track of the number of ADC interrupts over three AC cycles
// to calculate the average flyback current and PV panel voltage for MPPT

// To eliminate glitches at the ZC detection only look for a zero cross
// when inverterPeriodCounter is greater then 200
if((zeroCrossDetectFlag == 0) && (inverterPeriodCounter > 200))
{
zeroCrossDetection();
}

// Store current grid voltage to compare with next sample for ZCD
prevInverterOutputVoltage = inverterOutputVoltage;
prevInverterOutputVoltageFalse = inverterOutputVoltageFalse;



// Counter for verifying the Grid Frequency (number of ADC interrupts per grid half cycle)
// variable gets reset when finding the zero crossing event


inverterPeriodCounter++;
```

```
// Accumulate samples for the Average Calculation
inputCurrentSum = inputCurrentSum + flybackCurrent1 + flybackCurrent2;
inputVoltageSum = inputVoltageSum + pvPanelVoltage;


// ZeroCrossDetectFlag is set inside the zeroCrossDetection Routine
if(zeroCrossDetectFlag == 1)
{
zeroCrossDetectFlag = 0;
startupZeroCrossCounter++;

if ((systemState != SYSTEMSTARTUP)&&(systemState != ISLANDSTARTUP))
{
startupZeroCrossCounter = 0;
}

// Continuously Verify that Grid Frequency is okay
inverterFrequencyCheck();


// deltaAngle calculated based on present grid frequency/period measured
deltaAngle = __builtin_divsd((long)32767,inverterPeriod);

// Load the newly found Peak Voltage/Current of the Inverter Output (done @ zero crossing)
peakInverterOutputVoltage = maxInverterOutputVoltage;
peakInverterOutputCurrent = maxInverterOutputCurrent;
maxInverterOutputVoltage = 0;
maxInverterOutputCurrent = 0;

//Change ZeroCrossDelay based on peak AC Voltage
if(peakInverterOutputVoltage > 13800)
{
zeroCrossDelay = zeroCrossDelayMax;
}
else if (peakInverterOutputVoltage < 12000)
{
zeroCrossDelay = zeroCrossDelayMin;
}
else
{
zeroCrossDelay = zeroCrossDelayNom;
}


// Maximum MPPT calculation is based on the grid peak voltage
// mpptFacorMaximum = -(DERATINGSLOPE * peakInverterOutputVoltage) + DERATINGFACTOR
mpptFactorMaximum                                =                                DERATINGFACTOR-
((__builtin_mulss(peakInverterOutputVoltage,DERATINGSLOPE))>>14);


// Power de-rating routine ~7W per Volt when operating below 25Vdc
if((inputVoltageAverage<POWERDERATINGLIMIT)&&((systemState    ==    DAYMODE)||(systemState    ==
ISLANDMODE)))
{
// Determine MPPT Derating Factor (1V => 584 counts in Q15, 7W => 650 in Q15)
// Scale voltage properly
mpptDeratingFactor = POWERDERATINGLIMIT - inputVoltageAverage;
mpptDeratingFactor = ((__builtin_mulss(mpptDeratingFactor,DERATINGCONSTANT))>>14);

// Determine New MPPT Factor Maximum
mpptFactorMaximum = mpptFactorMaximum - mpptDeratingFactor;
}
```

```
peakOutputPower = ((__builtin_mulss(peakInverterOutputVoltage,peakInverterOutputCurrent))>> 14);

// Moving Average for peak power
peakOutputPowerSum=                                          peakOutputPowerSum+peakOutputPower-
peakOutputPowerArray[peakOutputPowerArrayCnt];
averagePeakOutputPower = peakOutputPowerSum >> 3;
peakOutputPowerArray[peakOutputPowerArrayCnt++] = peakOutputPower;

if(peakOutputPowerArrayCnt >= 8)
{
peakOutputPowerArrayCnt = 0;
}

// Burst Mode Detection (Output Power < ~15% Max Rated Power)
if((averagePeakOutputPower <= BURSTMODETHRESHOLDLOW) && (burstModeActiveFlag == 0))
{
burstModeActiveCounter++;

if(burstModeActiveCounter >= BURSTMODECOUNT)
{
burstModeActiveFlag = 1;
burstModeActiveCounter = 0;
mpptFactor = mpptFactor + (mpptFactor << 1);
}
}
else if ((burstModeActiveFlag == 1) && (averagePeakOutputPower > BURSTMODETHRESHOLDHIGH))
{
mpptFactor = (mpptFactor >> 1);
burstModeActiveFlag = 0;
}
else
{
burstModeActiveCounter = 0;
}

// Reset Variables every Zero Cross
sineAngle = 0;
globalAngle = 0;
ninetyDegreeDetectFlag = 0;
}



// This portion of SW determines the current reference from the sine lookup table based on
// deltaAngle which is calculated every Zero Cross
// sineAngle is used to point to specific location in sine lookup table
if(ninetyDegreeDetectFlag == 0)
{
// if sineAngle is less than 90 degrees, then add deltaAngle to point to next sample
if (sineAngle < NINETYDEGREE)
{
currentReferenceDynamic = sineTable512[((sineAngle) >> 5)];
sineAngle = sineAngle + deltaAngle;
}
else
{
sineAngle = NINETYDEGREE;
ninetyDegreeDetectFlag = 1;
}
}
else
{
// if sineAngle is greater than 90 degrees, then subtract deltaAngle from it
currentReferenceDynamic = sineTable512[((sineAngle) >> 5)];
sineAngle = sineAngle - deltaAngle;
}
```

```
// globalAngle is used for detecting grid voltage states
globalAngle = globalAngle + deltaAngle;

// startFullBridgeFlag enabled in system startup when several zero crosses have been
detected
// Flag is disabled when system enter fault mode
if(startFullBridgeFlag == 1)
{
// Routine that determines the state of the Full-Bridge MOSFETs
fullBridgeDrive();
}

// In Day mode Enable/Disbale the PWM drive for the flyback MOSFETs
if((systemState == DAYMODE)||(systemState == ISLANDMODE))
{
if ((fullBridgeState == FULLBRIDGE_Q3Q4_ACTIVE) || (fullBridgeState == FULLBRIDGE_Q2Q5_ACTIVE))
{
// If in Burst Mode, only deliver power for one complete AC cycle
// If not in Burst Mode, remove the PWM override when full-bridge state is active
if(burstModeActiveFlag == 0)
{
IOCON1bits.OVRENH = 0; // Remove Override
IOCON2bits.OVRENH = 0;
IOCON1bits.OVRENL = 0;
IOCON2bits.OVRENL = 0;

flybackControlLoop();
}
else if ((burstModeActiveFlag == 1) && (zeroCrossCount == 0))
{
IOCON1bits.OVRENH = 0; // Remove Override
IOCON2bits.OVRENH = 0;
IOCON1bits.OVRENL = 0;
IOCON2bits.OVRENL = 0;

flybackControlLoop();
}
}
else
{
IOCON1bits.OVRENH = 1; // Enable Override
IOCON1bits.OVRENL = 1;
IOCON2bits.OVRENH = 1;
IOCON2bits.OVRENL = 1;

// Give some history to the compensator before starting again at the zero cross
if(inverterPeriodMin == INVERTERPERIOD50HZMIN)
{
Ioutput = -1900;

if(peakInverterOutputVoltage > 13800)
{
Ioutput = -2500;
}
}
else
{
Ioutput = -500;//-1000;
}
}
}

//Can read AN10 and AN11 here data should be available at this time
driveSupplyVoltage = (ADCBUF11 << 5);
measuredTemperature = (ADCBUF10 << 5);
```

```
// Software for debugging purposes
#ifdef DMCI_ISR
if(dmciCount == 16)
{
array1 [dmciArrayIndex] = 0;
array2 [dmciArrayIndex] = 0;
array3 [dmciArrayIndex++] = 0;
if(dmciArrayIndex >= 100)
{
dmciArrayIndex = 0;
}
dmciCount = 0;
}

dmciCount++;

#endif

IFS7bits.ADCP3IF = 0; // Clear ADC Interrupt Flag
//LED_DRV3 = OFF;
}

void zeroCrossDetection(void)
{
//Actuas de una u otra forma dependiendo de en que estado estes

if((systemState!=ISLANDSTARTUP)&&(systemState != ISLANDMODE))
{
// Detect the zero crossing at the 1st Quadrant
if((prevInverterOutputVoltage < 0) && (inverterOutputVoltage >= 0))
{
// As a precaution change state of Full-Bridge if it hasn't already chaged
fullBridgeState = FULLBRIDGE_INACTIVE_4TH_QUADRANT;
ClrWdt();
zeroCrossDetectFlag = 1;
firstQuadrantFlag = 1;
// Allows Full-Bridge to change State
// Load counter to Grid Period to check grid frequency
inverterPeriod = inverterPeriodCounter;
inverterPeriodCounter = 0;
// Store accumulated voltage/current sum and counter for avg (avg calculated in T2ISR)
// The average is calculated every 3rd +ve zero cross event
zeroCrossCount++;
if (zeroCrossCount == 1)
{
numberofSamples = 0;
}
// inverterPeriod is only half cycle so add prevInverterPeriod
numberofSamples = numberofSamples + inverterPeriod + prevInverterPeriod;
if(zeroCrossCount >= 3)
{
zeroCrossCount = 0;
inputVoltageAverage = __builtin_divsd((long)inputVoltageSum ,(int)(numberofSamples));
inputCurrentAverage = __builtin_divsd((long)inputCurrentSum ,(int)(numberofSamples));
numberofSamples = 0;
inputVoltageSum = 0;
inputCurrentSum = 0;
avgInputDataReadyFlag = 1;



}
}
// Detect the zero crossing at the 3rd Quadrant
else if ((prevInverterOutputVoltage >= 0) && (inverterOutputVoltage < 0))
{
```

```
// As a precaution change state of Full-Bridge if it hasn't already chaged
fullBridgeState = FULLBRIDGE_INACTIVE_2ND_QUADRANT;
ClrWdt();
zeroCrossDetectFlag = 1;
thirdQuadrantFlag = 1; // Allows Full-Bridge to change State
// Load counter to Grid Period to check grid frequency
inverterPeriod = inverterPeriodCounter;
inverterPeriodCounter = 0;
}
}else //Si está en modo isla o empezando el modo isla
{
// Detect the zero crossing at the 1st Quadrant
if((prevInverterOutputVoltageFalse < 0) && (inverterOutputVoltageFalse >= 0))
{
// As a precaution change state of Full-Bridge if it hasn't already chaged
fullBridgeState = FULLBRIDGE_INACTIVE_4TH_QUADRANT;
ClrWdt();
zeroCrossDetectFlag = 1;
firstQuadrantFlag = 1; // Allows Full-Bridge to change State
// Grid Period Desired
//inverterPeriod = (inverterPeriodMin+inverterPeriodMax)/2;
inverterPeriod = inverterPeriodCounter;
inverterPeriodCounter = 0;
// Store accumulated voltage/current sum and counter for avg (avg calculated in T2ISR)
// The average is calculated every 3rd +ve zero cross event
zeroCrossCount++;
if (zeroCrossCount == 1)
{
numberofSamples = 0;
}
// inverterPeriod is only half cycle so add prevInverterPeriod
numberofSamples = numberofSamples + inverterPeriod + prevInverterPeriod;
if(zeroCrossCount >= 3)
{
zeroCrossCount = 0;
inputVoltageAverage = __builtin_divsd((long)inputVoltageSum ,(int)(numberofSamples));
inputCurrentAverage = __builtin_divsd((long)inputCurrentSum ,(int)(numberofSamples));
numberofSamples = 0;
inputVoltageSum = 0;
inputCurrentSum = 0;
avgInputDataReadyFlag = 1;
}
}
// Detect the zero crossing at the 3rd Quadrant
else if ((prevInverterOutputVoltageFalse >= 0) && (inverterOutputVoltageFalse < 0))
{
// As a precaution change state of Full-Bridge if it hasn't already chaged
fullBridgeState = FULLBRIDGE_INACTIVE_2ND_QUADRANT;
ClrWdt();
zeroCrossDetectFlag = 1;
thirdQuadrantFlag = 1; // Allows Full-Bridge to change State
// Grid Period Desired
//inverterPeriod = (inverterPeriodMin+inverterPeriodMax)/2;
inverterPeriod = inverterPeriodCounter;
inverterPeriodCounter = 0;
}
}
}

void inverterFrequencyCheck(void)
{
if ((inverterPeriod > inverterPeriodMin ) && (inverterPeriod < inverterPeriodMax))
{
if(inverterFrequencyErrorFlag == 1)
{
// Hysteresis for the frequency once a fault has been detected
if((inverterPeriod   >   (inverterPeriodMin   +   INVERTERPERIODHYS))   &&   (inverterPeriod   <
(inverterPeriodMax - INVERTERPERIODHYS)))
{
if(faultState == INVERTER_FREQUENCY)
```

```
{
faultState = NO_FAULT;
}

inverterFrequencyErrorFlag = 0;
}
}

frequencyFaultCounter = 0;
}
else
{
frequencyFaultCounter++;

if(frequencyFaultCounter >= 2)
{
inverterFrequencyErrorFlag = 1;
frequencyFaultCounter = 2;

if(faultState == NO_FAULT)
{
if((systemState != ISLANDSTARTUP)&&(systemState != ISLANDMODE))
{
faultState = INVERTER_FREQUENCY;
}
}
}
}

prevInverterPeriod = inverterPeriod; // Store Inverter Period for average calculation
}

void fullBridgeDrive(void)
{
if((systemState!= ISLANDSTARTUP)&&(systemState != ISLANDMODE))
{
        paso = PORTBbits.RB15;
}else
{
        if(thirdQuadrantFlag == 1){
                paso = 0;
        }
        if(firstQuadrantFlag ==1){
                paso =1;
        }
}
// Based on where we are in the sinewave enable/disable PWM3 Gate Drive (Full-Bridge)
switch(fullBridgeState)
{
case FULLBRIDGE_Q3Q4_ACTIVE:
{
if(globalAngle < NINETYDEGREE)
{
prevFullBridgeState = PORTBbits.RB15;
}

if (globalAngle > ONEHUNDREDSEVENTYFIVEDEGREE)
{
fullBridgeState = FULLBRIDGE_INACTIVE_2ND_QUADRANT;
}
}
break;
case FULLBRIDGE_INACTIVE_2ND_QUADRANT:
{
// Check for Hardware/Software Zero Cross
if((paso == 0) && (thirdQuadrantFlag == 1))
{
```

```
zcCounter++;

if(zcCounter >= zeroCrossDelay)
{
if((prevFullBridgeState==PORTBbits.RB15)&&((systemState==DAYMODE)||(systemState==ISLANDMODE)))
{
if(faultState == NO_FAULT)
{
faultState = HARDWAREZEROCROSS;
}
}

zcCounter = 0;
thirdQuadrantFlag = 0;
fullBridgeState = FULLBRIDGE_Q2Q5_ACTIVE;
}
}
else
{
zcCounter = 0;
break;
}
}
break;

case FULLBRIDGE_Q2Q5_ACTIVE:
{
if(globalAngle < NINETYDEGREE)
{
prevFullBridgeState = PORTBbits.RB15;
}

if (globalAngle > ONEHUNDREDSEVENTYFIVEDEGREE)
{
fullBridgeState = FULLBRIDGE_INACTIVE_4TH_QUADRANT;
}
}
break;

case FULLBRIDGE_INACTIVE_4TH_QUADRANT:
{
// Check for Hardware/Software Zero Cross
if ((paso == 1) && (firstQuadrantFlag == 1))
{
zcCounter++;

if(zcCounter >= zeroCrossDelay)
{
if((prevFullBridgeState==PORTBbits.RB15)&&((systemState==DAYMODE)||(systemState==ISLANDMODE)))
{
if(faultState == NO_FAULT)
{
faultState = HARDWAREZEROCROSS;
}
}

zcCounter = 0;
firstQuadrantFlag = 0;
fullBridgeState = FULLBRIDGE_Q3Q4_ACTIVE;
}


}
else
{
zcCounter = 0;
break;
```

```
}
}
break;
}

// Now that we know the state we can modify the PWM outputs

if(fullBridgeState == FULLBRIDGE_Q3Q4_ACTIVE)
{
OPTO_DRV1 = 1;
OPTO_DRV2 = 0;
IOCON3bits.OVRENL = 0;
}
else if (fullBridgeState == FULLBRIDGE_INACTIVE_2ND_QUADRANT)
{
IOCON3bits.OVRENH = 1;
IOCON3bits.OVRENL = 1;
OPTO_DRV2 = 1;
OPTO_DRV1 = 1;
}
else if (fullBridgeState == FULLBRIDGE_Q2Q5_ACTIVE)
{
OPTO_DRV2 = 1;
OPTO_DRV1 = 0;
IOCON3bits.OVRENH = 0;
}
else
{
IOCON3bits.OVRENH = 1;
IOCON3bits.OVRENL = 1;
OPTO_DRV2 = 1;
OPTO_DRV1 = 1;
}
}



void flybackControlLoop(void)
{
unsigned int flybackDutyCycle = 0;
unsigned int rectifiedVac = 0, decoupleTerm = 0;
int IoRef = 0, currentError = 0, Poutput = 0;
long int totalOutput = 0;
int peakVout=0;

// Compensator Software
IoRef = (__builtin_mulss((int)currentReferenceDynamic,(int)mpptFactor) >> 15);
currentError = IoRef - (rectifiedInverterOutputCurrent << 1);

Poutput = (__builtin_mulss(currentError,Ra)) >> 15;

Ioutput = Ioutput + (long)((__builtin_mulss((int)currentError,(int)Rsa)) >> 15);

if(Ioutput > 32767)
{
Ioutput = 32767;
}
else if(Ioutput < -32767)
{
Ioutput = -32767;
}
totalOutput = (long)Poutput + Ioutput;
//Get peak voltage from peakInverterOutputVoltage or from peakInverterOutputVoltage False

if((systemState != ISLANDSTARTUP)&&(systemState != ISLANDMODE)){
peakVout = peakInverterOutputVoltage;
}
```

```
else
{
peakVout = peakInverterOutputVoltageFalse;
}


// Input/Output Voltage Decoupling Software (Vo/(Vin + Vo))
// rectifiedVac is in Q13 format
rectifiedVac = (__builtin_mulss((int)currentReferenceDynamic,(int)peakVout) >> 16);

// pvPanel Voltage is multiplied by Q15(0.89) to have the same per unit as the rectifiedVac
// and both are in Q13 format (56.1 * turns ratio = 392.7, 392/445 = 0.89)
decoupleTerm = rectifiedVac + (__builtin_mulss(29163,(int)pvPanelVoltage) >> 17);

// Divide Q26/Q13, result is in Q13
decoupleTerm = __builtin_divsd(((long)rectifiedVac << 13) ,decoupleTerm);

totalOutput = totalOutput + (long)(decoupleTerm << 2);

// Clamp total output to the maximum Duty Cycle
if(totalOutput > MAXDUTYCLAMPED)
{
totalOutput = MAXDUTYCLAMPED;
}
else if(totalOutput < 0)
{
totalOutput = 0;
}

// Multiply flyback period by total output to get the flyback duty cycle
flybackDutyCycle = (__builtin_mulss((int)totalOutput,(int)FLYBACKPERIOD) >> 15);

// If duty cycle is less than dead-time make duty cycle equal to dead-time
if(flybackDutyCycle < FLYBACKALTDTR)
{
FLYBACKDUTY1 = FLYBACKALTDTR;
FLYBACKDUTY2 = FLYBACKALTDTR;
}
else
{
FLYBACKDUTY1 = flybackDutyCycle + deltaDutyCycle ;
FLYBACKDUTY2 = flybackDutyCycle - deltaDutyCycle ;
}

// Update ADC Triggers (trigger near end of PWM On-time)
TRIG1 = FLYBACKDUTY1 - (FLYBACKDUTY1 >> 2);
TRIG2 = FLYBACKDUTY2 - (FLYBACKDUTY2 >> 2);
}
```

Capítulo 5 STATEMACHINE.C

En esta sección se muestra el código del archivo statemachine.c, en él se realizan diversas comprobaciones de seguridad y se ejecuta la máquina de estados del código.

```
#include "Solar Microinverter_Statemachine.h"

// System Fault Variables
unsigned char storeFaultState = 0, faultCheckFlag = 0, criticalFaultRestartFlag = 0;
unsigned char pvPanelOverVoltageFlag = 0, pvPanelUnderVoltageFlag = 0, overTemperatureFlag
= 0;
unsigned char inverterOverVoltageFlag = 0, inverterUnderVoltageFlag = 0,
pvUnderVoltageCounter = 0;
unsigned char inverterUnderVoltageCounter = 0, inverterOverVoltageCounter = 0;
unsigned char tempFaultCnt = 0, driveSupplyFaultCnt = 0, referenceVoltageFaultCnt = 0;
unsigned int criticalFaultCounter = 0, pvPanelMPPOverVoltageCnt = 0;
unsigned int inverterOutputOverCurrent = 0;

// Start-up and Restart Variables
unsigned char zeroCrossDelay = 45, startFullBridgeFlag = 0;
unsigned char zeroCrossDelayNom = 30, zeroCrossDelayMin = 22, zeroCrossDelayMax = 37;
unsigned char acCurrentOffsetFlag = 0, acCurrentOffsetCounter = 0;
unsigned int systemRestartCounter = 0, acCurrentOffset = 0;
unsigned int inverterPeriodMin = INVERTERPERIOD60HZMIN, inverterPeriodMax =
INVERTERPERIOD50HZMAX;
long unsigned int acCurrentOffsetAverage = 0;

// MPPT and Load Balance Variables
unsigned char loadBalanceCounter = 0, mpptCounter = 0, mpptStartUpFlag = 0;
unsigned int inputVoltageAverage = 0, inputCurrentAverage = 0, openCircuitVoltage = 0;
unsigned int mpptFactor = MPPTFACTORMINIMUM;
int inputPower = 0, prevInputVoltageAverage = 0, deltaDutyCycle = 0;

// Startup in system error, If a fault is detected
// faultstate will change during the restart counter.
unsigned char systemState = SYSTEMERROR;
unsigned char switchState = SWITCHOFF;
unsigned char faultState = NO_FAULT;

// Externally Defined Variables
extern unsigned char avgInputDataReadyFlag, startupZeroCrossCounter;
extern unsigned char criticalFaultFlag, acCurrentOffsetFlag, ninetyDegreeDetectFlag;
extern unsigned char burstModeActiveFlag;
extern unsigned int mpptFactorMaximum, numberofSamples;
extern unsigned int measuredTemperature, driveSupplyVoltage, referenceVoltage;
extern unsigned int averageFlybackCurrent1, averageFlybackCurrent2;
extern unsigned int inverterPeriod, peakInverterOutputVoltage, burstModeActiveCounter;;
extern int inverterOutputCurrent;

// Variable Declaration for DMCI Debugging Tool
#ifdef DMCI_STATEMACHINE
int array1[100];
int array2[100];
int array3[100];
unsigned char dmciArrayIndex = 0;
#endif

#ifdef DMCI_MPPT
```

```
int array1[100];
int array2[100];
int array3[100];
unsigned char dmciArrayIndex = 0;
#endif

//Variables para el modo isla
long unsigned int ticks=0;

// Timer 3 Interrupt (300ms interrupt) for Fault indication
void __attribute__((__interrupt__, no_auto_psv)) _T3Interrupt()
{
static unsigned char ledCounter = 0, interruptCounter = 0;
if(ledCounter < storeFaultState)
{
LED_DRV1 ^= ON;
//Blink LED to indicate fault, storeFaultState is 2x to account for toggle Off
ledCounter++;
}
else if (ledCounter < (storeFaultState + 3)) // Wait three interrupts
{
ledCounter++;
}
else
{
ledCounter = 0;
}

if((systemState == DAYMODE)||(systemState == ISLANDMODE))
{
interruptCounter++;
}

// InterruptCounter is used to diplay the last known fault for some time after
// the system restarts (300ms * 200 ~ 1 minute)
if((faultState == NO_FAULT) && (interruptCounter >= 200))
{
T3CONbits.TON = 0;
//Disable this interrupt if the fault is removed and the delay has passed
ledCounter = 0;
interruptCounter = 0;
LED_DRV1 = OFF;
}

TMR3 = 0;
IFS0bits.T3IF = 0;
}



// Timer 1 Interrupt (51 us)
void __attribute__((__interrupt__, no_auto_psv)) _T1Interrupt()
{
TMR1 = 0;
//Ticks para el reloj del modo isla -> Tensión AC
ticks++;
if(ticks>=784) //80ms 1569 20ms 392 40ms 784
{
       ticks=0;
}
IFS0bits.T1IF = 0;
}



// Timer 2 Interrupt (100us interrupt), performs fault checking and system statemachine
void __attribute__((__interrupt__, no_auto_psv)) _T2Interrupt()
{
```

```
// Check Inverter Output Over Voltage Condition
if((peakInverterOutputVoltage > INVERTER_OVERVOLTAGE_LIMIT) && (inverterOverVoltageFlag ==
0))
{
inverterOverVoltageCounter++;

if(inverterOverVoltageCounter >= 2)
{
if(faultState == NO_FAULT)
{
faultState = INVERTER_VOLTAGE;
inverterOverVoltageFlag = 1;
}
}
}
else if((peakInverterOutputVoltage < INVERTER_OVERVOLTAGE_LIMIT_HYS) && (inverterOverVoltageFlag
== 1))
{
if(faultState == INVERTER_VOLTAGE)
{
faultState = NO_FAULT;
}
inverterOverVoltageFlag = 0;
}
else
{
inverterOverVoltageCounter = 0;
}

if((peakInverterOutputVoltage < INVERTER_UNDERVOLTAGE_LIMIT) && (inverterUnderVoltageFlag == 0))
{
inverterUnderVoltageCounter++;

if(inverterUnderVoltageCounter >= 2)
{
if(faultState == NO_FAULT)
{
if((systemState != ISLANDSTARTUP)&&(systemState != ISLANDMODE))
{
faultState = INVERTER_VOLTAGE;
inverterUnderVoltageFlag = 1;
}
}

inverterUnderVoltageCounter = 2;
}
}
else     if     ((peakInverterOutputVoltage     >     INVERTER_UNDERVOLTAGE_LIMIT_HYS)     &&
(inverterUnderVoltageFlag == 1))
{
if(faultState == INVERTER_VOLTAGE)
{
faultState = NO_FAULT;
}

inverterUnderVoltageFlag = 0;
}
else
{
inverterUnderVoltageCounter = 0;
}

// Only Check Input Voltage Fault When New Data is Available
if(avgInputDataReadyFlag == 1)
{
// Check PV Panel Voltage Using the Average Input Voltage
if((inputVoltageAverage > PVPANEL_OVERVOLTAGE_LIMIT) && (pvPanelOverVoltageFlag == 0))
{
```

```
if(faultState == NO_FAULT)
{
faultState = PV_PANEL_VOLTAGE;
pvPanelOverVoltageFlag = 1;
}
}
else if((inputVoltageAverage < PVPANEL_OVERVOLTAGE_LIMIT_HYS) && (pvPanelOverVoltageFlag == 1))
{
if(faultState == PV_PANEL_VOLTAGE)
{
faultState = NO_FAULT;
}

pvPanelOverVoltageFlag = 0;
}

// Check PV Panel Maximum Power Point Voltage Using the Average Input Voltage
if((inputVoltageAverage > PVPANEL_MPP_LIMIT) && ((systemState == DAYMODE)||(systemState ==
ISLANDMODE)))
{
pvPanelMPPOverVoltageCnt++;

// Allow time for system to find MPP
if(pvPanelMPPOverVoltageCnt > 1000)
{
pvPanelOverVoltageFlag = 1; // Set Flag so System Will Restart
pvPanelMPPOverVoltageCnt = 0;

if(faultState == NO_FAULT)
{
faultState = PV_PANEL_VOLTAGE;
}
}
}
else
{
pvPanelMPPOverVoltageCnt = 0;
}



// Check PV Panel Minimum Voltage Using Average Input Voltage
if((inputVoltageAverage < PVPANEL_UNDERVOLTAGE_LIMIT) && (pvPanelUnderVoltageFlag == 0))
{
pvUnderVoltageCounter++;

if(pvUnderVoltageCounter >= 10)
{
if(faultState == NO_FAULT)
{
faultState = PV_PANEL_VOLTAGE;
pvPanelUnderVoltageFlag = 1;
}

pvUnderVoltageCounter = 10;
}
}
else if ((inputVoltageAverage > PVPANEL_UNDERVOLTAGE_LIMIT_HYS) && (pvPanelUnderVoltageFlag ==
1))
{
if(faultState == PV_PANEL_VOLTAGE)
{
faultState = NO_FAULT;
}


pvPanelUnderVoltageFlag = 0;
```

```
}
else
{
pvUnderVoltageCounter = 0;
}
}

// Check Over Temperature Fault
if((measuredTemperature >= MAXTEMPERATURE) && (overTemperatureFlag == 0))
{
tempFaultCnt++;

if(tempFaultCnt > 100)
{
tempFaultCnt = 100;

if(faultState == NO_FAULT)
{
faultState = TEMPERATURE;
overTemperatureFlag = 1;
}
}
}
else if ((measuredTemperature <= MAXTEMPERATURE_HYS) && (overTemperatureFlag == 1))
{
overTemperatureFlag = 0;

if(faultState == TEMPERATURE)
{
faultState = NO_FAULT;
}
}
else
{
tempFaultCnt = 0;
}

//Check 12V Drive Supply Voltage
if((driveSupplyVoltage > MAXDRIVEVOLTAGE) || (driveSupplyVoltage < MINDRIVEVOLTAGE))
{
driveSupplyFaultCnt++;

if(driveSupplyFaultCnt > 10)
{
driveSupplyFaultCnt = 10;

if(faultState == NO_FAULT)
{
faultState = DRIVE_SUPPLY;
}
}
}
else
{
driveSupplyFaultCnt = 0;

if(faultState == DRIVE_SUPPLY)
{
faultState = NO_FAULT;
}
}

// Check Reference Voltage
if((referenceVoltage < MINREFERENCEVOLTAGE) || (referenceVoltage > MAXREFERENCEVOLTAGE))
{
referenceVoltageFaultCnt++;
```

```
if(referenceVoltageFaultCnt > 10)
{
referenceVoltageFaultCnt = 10;
if(faultState == NO_FAULT)
{
faultState = REFERENCE_VOLTAGE;
}
}
}
else
{
referenceVoltageFaultCnt = 0;

if (faultState == REFERENCE_VOLTAGE)
{
faultState = NO_FAULT;
}
}



// Check the fault state, if it is not equal to No Fault then set systemState = SYSTEM_ERROR
// Modified for isolated mode
if (faultState != NO_FAULT)
{
if((faultState == INVERTER_VOLTAGE)||(faultState == INVERTER_FREQUENCY)){
if((systemState != ISLANDSTARTUP)&&(systemState != ISLANDMODE)){
systemState = ISLANDSTARTUP;
faultState = NO_FAULT;
}
}else{
systemState = SYSTEMERROR;
}
}



// Check the On/Off switch state
if(PORTCbits.RC11 == 0)
{
switchState = SWITCHON;
}
else
{
//put system state in systemError
switchState = SWITCHOFF;
systemState = SYSTEMERROR;
}

switch(systemState)
{
case SYSTEMSTARTUP:
{
if(avgInputDataReadyFlag == 1)
{
// During system startup read the PV panel Voltage (open circuit voltage). This information
// is used to help speed up the time to find MPP
openCircuitVoltage = inputVoltageAverage;

avgInputDataReadyFlag = 0;
}

// Read AC Current offset during startup mode and verify data
// Needs to be completed before the full-bridge is enabled during system startup
if(acCurrentOffsetFlag == 0)
{
acCurrentOffsetAverage = acCurrentOffsetAverage + inverterOutputCurrent;
```

```
if(acCurrentOffsetCounter >= 255)
{
acCurrentOffset = (acCurrentOffsetAverage >> 8);
inverterOutputOverCurrent = INVERTER_OUTPUTCURRENT_MAX + (16383 - acCurrentOffset);
acCurrentOffsetFlag = 1;
acCurrentOffsetCounter = 0;
acCurrentOffsetAverage = 0;

if((acCurrentOffset < MINOFFSETCURRENT) || (acCurrentOffset > MAXOFFSETCURRENT))
{
if(faultState == NO_FAULT)
{
faultState = ACCURRENT_OFFSET;
}
}
}
else
{
acCurrentOffsetCounter++;
}
}

// At system Start-up, enable the Full-Bridge circuit (@ peak of AC Cycle) before the flyback
circuit is enabled
// If this is not done, the flyback output will have high DC voltage and when the full-bridge is
enabled at the
// zero cross there is a large dv/dt and the output current will have a large glitch and also
trip the flyback
// OVP circuit

if((startupZeroCrossCounter >= (ZEROCROSSCOUNT>>1)) && (ninetyDegreeDetectFlag == 1))
{
startFullBridgeFlag = 1;
// Set Flag to start full-bridge drive
}

// After several consecutive zero crossings switch to Day Mode
if(startupZeroCrossCounter > ZEROCROSSCOUNT)
{
if ((inverterPeriod > INVERTERPERIOD60HZMIN ) && (inverterPeriod <= INVERTERPERIOD60HZMAX))
{
inverterPeriodMin = INVERTERPERIOD60HZMIN;
inverterPeriodMax = INVERTERPERIOD60HZMAX;

// Delay at the zero crossings to allow AC voltage to reach flyback voltage
zeroCrossDelayNom = 20;
zeroCrossDelayMax = 22; // Delay slightly varies with AC voltage
zeroCrossDelayMin = 18;

zeroCrossDelay = zeroCrossDelayNom;
// Wait (17us*delay) at the zero cross before turning on flyback/full-bridge
}
else
{
inverterPeriodMin = INVERTERPERIOD50HZMIN;
inverterPeriodMax = INVERTERPERIOD50HZMAX;

// Delay at the zero crossings to allow AC voltage to reach flyback voltage
zeroCrossDelayNom = 30;
zeroCrossDelayMax = 35;
// Delay slightly varies with AC voltage
zeroCrossDelayMin = 25;

zeroCrossDelay = zeroCrossDelayNom;
// Wait (17us*delay) at the zero cross before turning on flyback/full-bridge
```

```
}

// Change system state to Day Mode
systemState = DAYMODE;
}
}
break;

case DAYMODE:
{
// When average input voltage and average input current are available call MPPT Routine
if(avgInputDataReadyFlag == 1)
{
#ifndef BENCHTESTING
MPPTRoutine(); // Call MPPT Routine when data is ready
#endif

// Change Current Reference based on the PV panel voltage
if(inputVoltageAverage >= PVPANEL_40V)
{
CMPDAC2bits.CMREF = 700; // Consider current at ~39V
CMPDAC3bits.CMREF = 700;
}
else if(inputVoltageAverage >= PVPANEL_30V)
{
CMPDAC2bits.CMREF = 850; // Consider current at ~29V
CMPDAC3bits.CMREF = 850;
}
else
{
CMPDAC2bits.CMREF = 1000; // Consider current at ~20V
CMPDAC3bits.CMREF = 1000;
}

avgInputDataReadyFlag = 0;
}

// Load Balance routine to make sure that both flyback converters are sharing the load equally
~50%
// Execute at a slower rate 100us * LOADBALCOUNT

if(loadBalanceCounter >= LOADBALCOUNT)
{
LoadBalance(); // Call Load Balance Routine
loadBalanceCounter = 0;
}

loadBalanceCounter++;

// Software for soft-start when using a bench supply as MPP-Tracking is removed
#ifdef BENCHTESTING

mpptCounter++;

if(mpptFactor < MPPTFACTOR_BENCHTESTING)
{
if(mpptCounter >= MPPTCOUNT)
{
mpptFactor = mpptFactor + MPPTFACTORINCREMENT;
mpptCounter = 0;
}
}
else
{
mpptFactor = MPPTFACTOR_BENCHTESTING;
```

```
mpptCounter = 0;
}

#endif

}
break;

case SYSTEMERROR:
{
IOCON1bits.OVRENH = 1;
IOCON1bits.OVRENL = 1;
IOCON2bits.OVRENH = 1;
IOCON2bits.OVRENL = 1;
IOCON3bits.OVRENH = 1;
IOCON3bits.OVRENL = 1;
OPTO_DRV1 = 0;
OPTO_DRV2 = 0;
mpptStartUpFlag = 0;
startFullBridgeFlag = 0; // Reset flag to start full-bridge
burstModeActiveCounter = 0; // Reset burst mode
burstModeActiveFlag = 0;
acCurrentOffset = 0;
acCurrentOffsetFlag = 0; // Allow system to re-calculate AC Current Offset
mpptFactor = MPPTFACTORMINIMUM;
inputPower = 0;
prevInputVoltageAverage = 0;

ClrWdt(); // In fault mode we need to clear WDT

if((switchState == SWITCHON) && (faultState == NO_FAULT))
{
// Switch to system startup after no faults have been detected for ~1s

systemRestartCounter++;

if(systemRestartCounter >= RESTARTCOUNT)
{
systemState = SYSTEMSTARTUP;
systemRestartCounter = 0;
}
}
else if((switchState == SWITCHON) && (faultState != NO_FAULT))
{
// If a fault is present then diplay the fault using T3 and LED D27 on the PCB
systemRestartCounter = 0;
storeFaultState = (faultState << 1); // x2 to account for "off" time
T3CONbits.TON = 1;

// Remove the fault and allow system to try to restart
// If the fault is ac current offset or HW Zero Cross
if((faultState == ACCURRENT_OFFSET) || (faultState == HARDWAREZEROCROSS))
{
faultState = NO_FAULT;
}

// Critical Faults: AC Current, Flyback Over Voltage, and Flyback Over Current
// Handle faults differently: Allow system to try to restart only once, if fault
// is still present then disable PWM module.
if((criticalFaultFlag == 1) && (criticalFaultRestartFlag == 0))
{
criticalFaultCounter++;

// After 2s remove the critical fault and allow system to try to restart
if(criticalFaultCounter > CRITICALFAULTCOUNT)
```

```
{
criticalFaultRestartFlag = 1;

// As the fault for flyback over current is latched the PWM needs to
// exit the latched fault mode in order to restart
if(faultState == FLYBACK_OVERCURRENT)
{
FCLCON1bits.FLTMOD = 3; // Disable Fault Mode
FCLCON2bits.FLTMOD = 3; // Disable Fault Mode
FCLCON1bits.FLTMOD = 0; // Latched Fault Mode
FCLCON2bits.FLTMOD = 0; // Latched Fault Mode
}

faultState = NO_FAULT;
criticalFaultCounter = 0;
}
}
}
else if (switchState == SWITCHOFF)
{
// Reset Period Limits if switch is off
// This is only required for testing 50/60 Hz operation
// without having to cycle power
inverterPeriodMin = INVERTERPERIOD60HZMIN;
inverterPeriodMax = INVERTERPERIOD50HZMAX;
T3CONbits.TON = 0;
LED_DRV1 = OFF;
}


}
break;

//NUEVOS ESTADOS PARA EL MODO ISLA

case ISLANDSTARTUP:
{
ClrWdt(); // In this mode we need to clear WDT

inverterPeriodMin = INVERTERPERIOD50HZMIN;
inverterPeriodMax = INVERTERPERIOD50HZMAX;
//inverterPeriod = (inverterPeriodMin+inverterPeriodMax)/2;
zeroCrossDelayNom = 30;
zeroCrossDelayMax = 35; // Delay slightly varies with AC voltage
zeroCrossDelayMin = 25;
zeroCrossDelay = zeroCrossDelayNom;

if(avgInputDataReadyFlag == 1)
{
// During system startup read the PV panel Voltage (open circuit voltage). This information
// is used to help speed up the time to find MPP
openCircuitVoltage = inputVoltageAverage;

avgInputDataReadyFlag = 0;
}

// Read AC Current offset during startup mode and verify data
// Needs to be completed before the full-bridge is enabled during system startup
if(acCurrentOffsetFlag == 0)
{
acCurrentOffsetAverage = acCurrentOffsetAverage + inverterOutputCurrent;

if(acCurrentOffsetCounter >= 255)
{
acCurrentOffset = (acCurrentOffsetAverage >> 8);
inverterOutputOverCurrent = INVERTER_OUTPUTCURRENT_MAX + (16383 - acCurrentOffset);
acCurrentOffsetFlag = 1;
```

```
acCurrentOffsetCounter = 0;
acCurrentOffsetAverage = 0;

if((acCurrentOffset < MINOFFSETCURRENT) || (acCurrentOffset > MAXOFFSETCURRENT))
{
if(faultState == NO_FAULT)
{
faultState = ACCURRENT_OFFSET;
}
}
}
else
{
acCurrentOffsetCounter++;
}
}

startFullBridgeFlag = 1; // Set Flag to start full-bridge drive

// After several consecutive zero crossings switch to Day Mode
if(startupZeroCrossCounter > ZEROCROSSCOUNT)
{
if ((inverterPeriod > INVERTERPERIOD60HZMIN ) && (inverterPeriod <= INVERTERPERIOD60HZMAX))
{
inverterPeriodMin = INVERTERPERIOD60HZMIN;
inverterPeriodMax = INVERTERPERIOD60HZMAX;

// Delay at the zero crossings to allow AC voltage to reach flyback voltage
zeroCrossDelayNom = 20;
zeroCrossDelayMax = 22; // Delay slightly varies with AC voltage
zeroCrossDelayMin = 18;

zeroCrossDelay = zeroCrossDelayNom;
// Wait (17us*delay) at the zero cross before turning on flyback/full-bridge
}
else
{
inverterPeriodMin = INVERTERPERIOD50HZMIN;
inverterPeriodMax = INVERTERPERIOD50HZMAX;

// Delay at the zero crossings to allow AC voltage to reach flyback voltage
zeroCrossDelayNom = 30;
zeroCrossDelayMax = 35; // Delay slightly varies with AC voltage
zeroCrossDelayMin = 25;

zeroCrossDelay = zeroCrossDelayNom;
// Wait (17us*delay) at the zero cross before turning on flyback/full-bridge
}

// Change system state to Island Mode
systemState = ISLANDMODE;
}
}
break;

case ISLANDMODE:
{
ClrWdt(); // In this mode we need to clear WDT

// When average input voltage and average input current are available call MPPT Routine
if(avgInputDataReadyFlag == 1)
{
#ifndef BENCHTESTING
MPPTRoutine(); // Call MPPT Routine when data is ready
#endif
```

```
// Change Current Reference based on the PV panel voltage
if(inputVoltageAverage >= PVPANEL_40V)
{
CMPDAC2bits.CMREF = 700; // Consider current at ~39V
CMPDAC3bits.CMREF = 700;
}
else if(inputVoltageAverage >= PVPANEL_30V)
{
CMPDAC2bits.CMREF = 850; // Consider current at ~29V
CMPDAC3bits.CMREF = 850;
}
else
{
CMPDAC2bits.CMREF = 1000; // Consider current at ~20V
CMPDAC3bits.CMREF = 1000;
}

avgInputDataReadyFlag = 0;
}

// Load Balance routine to make sure that both flyback converters are sharing the load equally
~50%
// Execute at a slower rate 100us * LOADBALCOUNT

if(loadBalanceCounter >= LOADBALCOUNT)
{
LoadBalance(); // Call Load Balance Routine
loadBalanceCounter = 0;
}

loadBalanceCounter++;

// Software for soft-start when using a bench supply as MPP-Tracking is removed
#ifdef BENCHTESTING

mpptCounter++;

if(mpptFactor < MPPTFACTOR_BENCHTESTING)
{
if(mpptCounter >= MPPTCOUNT)
{
mpptFactor = mpptFactor + MPPTFACTORINCREMENT;
mpptCounter = 0;
}
}
else
{
mpptFactor = MPPTFACTOR_BENCHTESTING;
mpptCounter = 0;
}

#endif

}
break;
}



// Software for debugging purposes
#ifdef DMCI_STATEMACHINE
array1 [dmciArrayIndex] = deltaDutyCycle;
array2 [dmciArrayIndex] = averageFlybackCurrent2 - averageFlybackCurrent1;
array3 [dmciArrayIndex++] = 0;
```

```
if(dmciArrayIndex >= 100)
{
dmciArrayIndex = 0;
}
#endif

TMR2 = 0;
IFS0bits.T2IF = 0;
}



// This MPPT algorithim implements the Perturbation and Observation method for detecting
Maximum
// Power Point. MPPT can be up to ~25000
void MPPTRoutine(void)
{
int deltaV = 0, prevInputPower = 0;
unsigned char mpptScaleFactor = 0;


//Store off previous inputPower
prevInputPower = inputPower;


// Calculate new input power and change in input voltage
inputPower = (__builtin_mulss((int)inputVoltageAverage ,(int)inputCurrentAverage) >> 15);
deltaV = inputVoltageAverage - prevInputVoltageAverage;


// If there is a large drop in voltage decrease mpptFactor at a faster rate
// Example would be large AC voltage fluctuations
if(deltaV <= PVPANEL_VOLTAGEDROP)
{
mpptFactor = mpptFactor - (mpptFactor >> 2); // Reset to 75% power
}


// To find MPP faster, increase mpptFactor at a faster rate until the operating voltage
// is less than the opencircuit voltage minus ~3V. Vmp and Voc should always be more
// than 5-6V difference at any operating temperature or irradiance.
if((inputVoltageAverage > (openCircuitVoltage - 1650)) && (mpptStartUpFlag == 0))
{
mpptFactor += 50;
}
else
{
mpptStartUpFlag = 1;
criticalFaultFlag = 0;
criticalFaultRestartFlag = 0;
}



if(burstModeActiveFlag == 1)
{
mpptScaleFactor = 1;
}
else
{
mpptScaleFactor = 0;
}

if (inputPower > prevInputPower)
{
if (deltaV < -LARGEVOLTAGEDIFFERENCE)
{
mpptFactor += MININCREMENTMPPTFACTOR;
}
else if (deltaV < 0)
{
```

```
mpptFactor += MAXINCREMENTMPPTFACTOR;
}
else if (deltaV > LARGEVOLTAGEDIFFERENCE)
{
mpptFactor -= (MAXDECREMENTMPPTFACTOR << mpptScaleFactor);
}
else if (deltaV > 0)
{
mpptFactor -= (MINDECREMENTMPPTFACTOR << mpptScaleFactor);
}
}
else if (inputPower < prevInputPower)
{
if (deltaV < -LARGEVOLTAGEDIFFERENCE)
{
mpptFactor -= (MAXDECREMENTMPPTFACTOR << mpptScaleFactor);
}
else if (deltaV < 0)
{
mpptFactor -= (MINDECREMENTMPPTFACTOR << mpptScaleFactor);
}
else if (deltaV > LARGEVOLTAGEDIFFERENCE)
{
mpptFactor += MAXINCREMENTMPPTFACTOR;
}
else if (deltaV > 0)
{
mpptFactor += MININCREMENTMPPTFACTOR;
}
}

// Saturate the MPPT limit to min and max values
if(mpptFactor > mpptFactorMaximum)
{
mpptFactor = mpptFactorMaximum;
}
else if(mpptFactor < MPPTFACTORMINIMUM)
{
mpptFactor = MPPTFACTORMINIMUM;
}

// Store off last known input power and input voltage
prevInputVoltageAverage = inputVoltageAverage;

// Software for debugging purposes
#ifdef DMCI_MPPT
array1 [dmciArrayIndex] = mpptFactor;
array2 [dmciArrayIndex] = deltaV;
array3 [dmciArrayIndex++] = inputPower;
if(dmciArrayIndex >= 100)
{
dmciArrayIndex = 0;
}
#endif
}



void LoadBalance(void)
{
static int loadBalIoutput = 0;
int diffFlybackCurrent = 0, loadBalPoutput = 0, loadBalPIoutput = 0;

// Difference of the two Flyback MOSFET Currents
diffFlybackCurrent = averageFlybackCurrent2 - averageFlybackCurrent1;


// Error * Proportional Gain
```

```
loadBalPoutput = ( (__builtin_mulss(diffFlybackCurrent,(int)KAQ15)) >> 15);

// Error * Integral Gain
loadBalIoutput = loadBalIoutput + ( (__builtin_mulss(diffFlybackCurrent,(int)KSAQ15)) >> 15);

// Check for Integral term exceeding MAXBALANCE, If true, saturate the integral term to
MAXBALANCE
// Check for Integral term going below -MAXBALANCE, If true, saturate the integral term to -
MAXBALANCE
if(loadBalIoutput > MAXBALANCE)
{
loadBalIoutput = MAXBALANCE;
}
else if(loadBalIoutput < -MAXBALANCE)
{
loadBalIoutput = -MAXBALANCE;
}

// PI Output = Proportional Term + Integral Term
loadBalPIoutput = loadBalPoutput + loadBalIoutput;

// Check for PI Output exceeding MAXBALANCE, If true, saturate PI Output to MAXBALANCE
// Check for PI Output going below -MAXBALANCE, If true, saturate PI Output to -MAXBALANCE
if(loadBalPIoutput > MAXBALANCE)
{
loadBalPIoutput = MAXBALANCE;
}
else if(loadBalPIoutput < -MAXBALANCE)
{
loadBalPIoutput = -MAXBALANCE;
}
```

DOCUMENTO II

PRESUPUESTO

Índice

1. Mediciones …… 148

1.1. Componentes principales……..... 148
1.2. Equipo y herramientas……..... 151
1.3. Software……..... 151
1.4. Mano de obra directa…… 152
1.5. Servicios Auxiliares……. 152

2. Precios unitarios …… 153

2.1. Componentes principales……..... 153
2.2. Equipo y herramientas……..... 156
2.3. Software……..... 156
2.4. Mano de obra directa……. 157
2.5. Servicios Auxiliares……... 157

3. Sumas parciales ……..…….…… 158

3.1. Componentes principales……..... 158
3.2. Equipo y herramientas……..... 162
3.3. Software……..... 163
3.4. Mano de obra directa……..……... 163
3.5. Servicios Auxiliares……..…… 163

4. Presupuesto general ……..…….. 164

Capítulo 1

Mediciones

En este capítulo se recogen las unidades de los componentes y equipos utilizados, así como las horas imputadas al proyecto, divididas en varias partidas diferentes.

1.1. Componentes principales

Nombre	Componentes	Uds
C1	Condensador capacitivo cerámico 1.0UF, 100V X7R 1206	1
C2	Condensador capacitivo cerámico 10uF, 100V X7R 10% 0805	1
C3, C32, C44, C45, C46, C47, C80, C81	Condensador capacitivo cerámico 0.01uF, 50V X7R	8
C4, C7, C8, C9, C11, C14, C21, C22, C23, C24, C28, C35, C40, C54, C57, C74, C75, C77, C78, C82, C89	Condensador capacitivo cerámico 0.1uF, 50V X7R	21
C5, C20, C25, C29, C39, C55, C76, C79	Condensador capacitivo cerámico 1uF, 16V X5R	8
C6	Condensador electrolítico de Tántalo 100uF, 16V 10% 2917	1
C10, C13, C16, C17, C18, C19	Condensador capacitivo cerámico 10uF, 16V X5R	6
C27, C33, C37	Condensador capacitivo cerámico 8200pF, 10% 50V X7R 0603	3
C30	Condensador cerámico multicapa 2200pF, 0603 50V X7R 20%	1
C41, C42, C43	Condensador cerámico multicapa 470pF, 50V CERAMIC X7R 0603	3
C48, C49, C52, C53	Condensador capacitivo Cerámico 4700pF, 250VAC X1Y2 Rad	4
C87	Condensador capacitivo Cerámico 4700pF, 250VAC X1Y2 Rad	1
C50, C51, C88	Condensador de película MMKP 0.015uF, 1000VDC 5%	3
C56	Condensador cerámico multicapa 1000pF, 50V X7R 0603	1
C58, C69	Condensador de película 0.1uF, 630VDC RADIAL	2
C59, C70	Condensador capacitivo Cerámico 2200pF, 5% 100V NP0 1210	2
C60, C71	Condensador capacitivo Cerámico 10uF, 100V X7S 2220	2
C61, C62, C72, C73, C83, C84, C85, C86	Condensador capacitivo Cerámico 2.2uF, 250V 20% X7T 2225	8
C63, C64, C65, C66, C67	Condensador electrolítico de Aluminio 2200uF, 63V 20% Radial	5
C68	Condensador capacitivo Cerámico 120pF, 50V 5% C0G 0603	1
C90	Condensador capacitivo Cerámico 220pF, 50V 10% X7R 0603	1
D1	Diodo Schottky 1A, 60V SMA	1
D2	Diodo switch 75V 300mA, SOD323	1
D3, D9, D11	Diodo Schottky 500mA, 20V 0603	3
D4, D26, D27	LED Rojo, SMD 0603	3
D5	LED Verde, SMD 0603	1

Nombre	Componentes	Uds
D6	LED Amarillo SMD 0603	1
D7, D8, D10, D12, D28	Diodo Schottky 200mA, 30V SOT23	5
D13	Referencia tensión 2,5V 1% Aj. SOT23F	1
D14, D15, D16, D17, D20, D21, D22, D23, D24, D25	Diodo Schottky 500MA, 40V SOD123	10
D18, D19	Diodo Schottky 10A, 1200V TO252-2	2
D29	Diodo Zener TVS Bidireccional 400W 10V SMA	1
F1, F2	Fusible Lento 250VAC 6.3A Radial	2
J1	Conector modular Jack 6-6 RT/A PCB 50AU	1
J2	Conector Terminal Strip, 2X6, 0.100sp, 0.025 sq post	1
J3	Conector Terminal Strip, 1X2, 0.100sp, 0.025 sq post	1
J4	Conector con barreras Strip 3POS .375"	1
J5, J6	Conector PCB .250	2
L1	Bobina 220uH, 700 mOhms DCR 5MHz	1
L2	Bobina 22uH, 787 mOhms DCR 29.4 MHz	1
L3	Bobina 10uH 10%	1
L4, L7	Bobina toroidal 150UH 15% HORIZ	2
L5, L8	Núcleo de ferrita para fusibles 60 OHM 6000MA 1806	2
L6	Filtro de modo común 3.3uH, 2.5A	1
L9	Inductor fijo 0.22uH 40.5A SMD	1
MOV1, MOV2	Varistor 275V RMS 14mm Radial	2
Q1	Transistor bipolar NPN GP 500mA 45V SOT23	1
Q2, Q3, Q4, Q5	Transistor Mosfet Canal N 600V 20.2A TO263	4
Q6, Q8	Transistor Mosfet Canal P 150V 8.9A 1212-8	2
Q7, Q9	Transistor Mosfet Canal N 150V 83A D2PAK	2
R1, R10, R52, R57, R64, R69, R78, R80, R86, R88, R93, R94, R104, R107, R110, R111, R118, R119, R126, R127	Resistencia 10 KOhms 1/10W 1% 0603 SMD	20
R2, R32, R37, R38, R39, R41, R42, R44, R45	Resistencia 3.3 KOhms 1/10W 1%	9
R3	Resistencia 165 KOhms 1/10W 1% 0603 SMD	1
R4	Resistencia 10.2 KOhms 1/10W 1% 0603 SMD	1
R5	Resistencia 1.5 Ohms 1/4W 5% 1206 SMD	1
R6	Resistencia 169K Ohms 1/10W 1% 0603 SMD	1
R7	Resistencia 2.70K Ohms 1/10W 1% 0603 SMD	1
R8, R133, R134, R137, R138	Resistencia 0.0 Ohms 1/4W 1206 SMD	5
R9	Resistencia 52.3 KOhms 1/10W 1% 0603 SMD	1
R13, R22, R43, R47, R56, R59, R68, R70, R112, R117, R120, R125, R135, R136	Resistencia 0.0 Ohms 1/10W 0603 SMD	14

Nombre	Componentes	Uds
R14	Resistencia 4.70 KOhms 1/10W 1% 0603 SMD	1
R15	Resistencia 470 Ohms 1/10W 1% 0603 SMD	1
R16, R17, R18, R71, R129, R130	Resistencia 330 Ohms 1/10W 1%	6
R19, R25, R28, R34	Resistencia, 100 KOhms 1/10W 1%	4
R20, R27	Resistencia 1.60 KOhms 1/10W 1% 0603 SMD	2
R21, R106	Resistencia 5.10 KOhms 1/10W 1% 0603 SMD	2
R23, R36	Resistencia 6.2 KOhms 1/10W 5% 0603 SMD	2
R24, R31	Resistencia 27 KOhms 1/10W 5% 0603 SMD	2
R26, R30, R128	Resistencia 1.69 KOhms 1/10W 1% 0603 SMD	3
R29, R73, R91, R92, R113, R115, R121, R123	Resistencia 1 KOhms 1/10W 1% 0603 SMD	8
R35	Resistencia 2.4 KOhms 1/10W 5% 0603 SMD	1
R40	Resistencia 470 KOhms 1/10W 1%	1
R46, R55, R58, R67, R74	Resistencia 7.5 KOhms 1/10W 1% 0603 SMD	5
R48, R50, R60, R62	Resistencia 3.01 KOhms 1/10W 1%	4
R49, R61	Resistencia 120 Ohms 1/10W 1%	2
R54, R66	Resistencia 15.0 Ohms 1/10W 1% 0603 SMD	2
R72	Resistencia 120 KOhms 1/10W 1% 0603 SMD	1
R75, R76, R83, R84, R95, R96, R114, R116, R122, R124	Resistencia 11.0 Ohms 1/10W 1% 0603 SMD	10
R77, R79, R85, R87	Resistencia 1.0 Ohms 1/10W 5% 0603 SMD	4
R81, R82, R89, R90	Resistencia 270 Ohms 1/10W 1% 0603 SMD	4
R97, R99, R101	Resistencia 150 KOhms 1/4W 5% 1206 SMD	3
R98, R100	Resistencia 240 KOhms 1/4W 5% 1206 SMD	2
R102	Resistencia 300 KOhms 1/4W 5% 1206 SMD	1
R103	Resistencia 100 Ohms 1/10W 5% 0603 SMD	1
R105	Resistencia 30 KOhms 1/10W 5% 0603 SMD	1
R108	Resistencia 20.0 KOhms 1/10W 1% 0603 SMD	1
R109	Resistencia 15.0 KOhms 1/10W 1% 0603 SMD	1
R141	Resistencia 3.3 KOhms 1/10W 1% 0603 SMD	1
SW1	Interruptor táctil SPST-NO 0.05A 12V	1
TP1, TP2, TP3, TP4, TP5, TP6, TP7, TP8, TP9, TP10, TP11, TP12, TP13, TP14, TP15, TP16, TP17, TP18, TP19, TP20, TP21, TP22, TP23, TP24, TP25, TP26, TP27, TP28, TP29, TP30, TP31	Utillaje puntos de medida	31
TR1	Transformador 1VA, 115/230V 10VCT 5V 10VCT 90mA	1
TR2, TR3	Transformador 264uH 1.5 DCR 3.6MH, 95pF	2
TR4, TR6	Núcleo de transformador RM14 230Vac	2
Base de TR4 y TR6	Soporte núcleo bobina Coilformer RM14 12-pins	2

Nombre	Componentes	Uds
TR5, TR7	Transformador del sensor de corriente3000uH SMD	2
U1	Microprocesador High-Performance 32-bit Digital Signal Controller, 44-Pin QFN	1
U2	Regulador de conmutación IC BUCK ADJ .35A 8MSOP	1
U3	Convertidor de tensión IC REG BUCK ADJ 0.6A SOT23-6	1
U4	Regulador de tensión 3.3V 250MA SOT-89-3	1
U5, U6, U7	Amplificador operacional dual 2.5V 8SOIC	3
U8	Sensor de temperature lineal SC70-5	1
U9, U10, U11, U12	Optoacoplador para salida de transistores 4-SMD	4
U13, U16, U17	Driver IC MOSFET DVR 4.0A DUAL 8SOIC	3
U14	Sensor de corriente de efecto hall 5A 5V 8-SOIC	1
U15	Amplificador de error con aislamiento óptico 8-SMD	1
SW2	Interruptor ON/OFF 10A 250V	1
J3	Conector hembra para PCB .100"	1
CIR	Circuito impreso	1

1.2. Equipos y herramientas

Elemento	Cantidad	Horas de proyecto	Horas de uso al año
Ordenador	1	500	1600
Programador MPLAB ICD3	1	200	500
Fuente alimentación 30V 5A	1	200	400
Osciloscopio 4 canales	1	20	300
Polímetro	1	50	250
Destornillador, alicates y otras herramientas	1	20	400
Regleta de protección	1	500	1600

1.3. Software

Programa	Cantidad	Horas de proyecto	Horas de uso al año
Microsoft Word	1	100	500
Microsoft Excel	1	150	500
Eagle Profesional 6.1	1	100	500
MPLAB v8.85	1	350	500

1.4. Mano de obra directa

Actividad	Horas
Diseño de tarjeta	100
Búsqueda de componentes	100
Programación	200
Pruebas y solución de problemas	150
Documentación del proyecto	100
Horas totales	650

1.5. Servicios auxiliares

Actividad	Unidades
Proceso de fabricación	1

Capítulo 2

Precios unitarios

En este capítulo se indican los precios de cada uno de los elementos necesarios para llevar a cabo este proyecto, los cuales se han citado anteriormente.

2.1. Componentes principales

Nombre	Componentes	Precio (€/ud.)
C1	Condensador capacitivo cerámico 1.0UF, 100V X7R 1206	0,282
C2	Condensador capacitivo cerámico 10uF, 100V X7R 10% 0805	0,07
C3, C32, C44, C45, C46, C47, C80, C81	Condensador capacitivo cerámico 0.01uF, 50V X7R	0,006
C4, C7, C8, C9, C11, C14, C21, C22, C23, C24, C28, C35, C40, C54, C57, C74, C75, C77, C78, C82, C89	Condensador capacitivo cerámico 0.1uF, 50V X7R	0,014
C5, C20, C25, C29, C39, C55, C76, C79	Condensador capacitivo cerámico 1uF, 16V X5R	0,062
C6	Condensador electrolítico de Tántalo 100uF, 16V 10% 2917	1,004
C10, C13, C16, C17, C18, C19	Condensador capacitivo cerámico 10uF, 16V X5R	0,21
C27, C33, C37	Condensador capacitivo cerámico 8200pF, 10% 50V X7R 0603	0,096
C30	Condensador cerámico multicapa 2200pF, 0603 50V X7R 20%	0,444
C41, C42, C43	Condensador cerámico multicapa 470pF, 50V CERAMIC X7R 0603	0,033
C48, C49, C52, C53	Condensador capacitivo Cerámico 4700pF, 250VAC X1Y2 Rad	0,592
C87	Condensador capacitivo Cerámico 4700pF, 250VAC X1Y2 Rad	0,592
C50, C51, C88	Condensador de película MMKP 0.015uF, 1000VDC 5%	0,76
C56	Condensador cerámico multicapa 1000pF, 50V X7R 0603	0,192
C58, C69	Condensador de película 0.1uF, 630VDC RADIAL	0,337
C59, C70	Condensador capacitivo Cerámico 2200pF, 5% 100V NP0 1210	0,568
C60, C71	Condensador capacitivo Cerámico 10uF, 100V X7S 2220	1,766
C61, C62, C72, C73, C83, C84, C85, C86	Condensador capacitivo Cerámico 2.2uF, 250V 20% X7T 2225	1,73
C63, C64, C65, C66, C67	Condensador electrolítico de Aluminio 2200uF, 63V 20% Radial	2,3
C68	Condensador capacitivo Cerámico 120pF, 50V 5% C0G 0603	0,071
C90	Condensador capacitivo Cerámico 220pF, 50V 10% X7R 0603	0,039
D1	Diodo Schottky 1A, 60V SMA	0,104
D2	Diodo switch 75V 300mA, SOD323	0,163
D3, D9, D11	Diodo Schottky 500mA, 20V 0603	0,352
D4, D26, D27	LED Rojo, SMD 0603	0,104
D5	LED Verde, SMD 0603	0,112

Nombre	Componentes	Precio (€/ud.)
D6	LED Amarillo SMD 0603	0,112
D7, D8, D10, D12, D28	Diodo Schottky 200mA, 30V SOT23	0,143
D13	Referencia tensión 2,5V 1% Aj. SOT23F	0,124
D14, D15, D16, D17, D20, D21, D22, D23, D24, D25	Diodo Schottky 500MA, 40V SOD123	0,192
D18, D19	Diodo Schottky 10A, 1200V TO252-2	6,4
D29	Diodo Zener TVS Bidireccional 400W 10V SMA	0,78
F1, F2	Fusible Lento 250VAC 6.3A Radial	0,663
J1	Conector modular Jack 6-6 RT/A PCB 50AU	0,97
J2	Conector Terminal Strip, 2X6, 0.100sp, 0.025 sq post	1,48
J3	Conector Terminal Strip, 1X2, 0.100sp, 0.025 sq post	0,325
J4	Conector con barreras Strip 3POS .375"	3,18
J5, J6	Conector PCB .250	0,068
L1	Bobina 220uH, 700 mOhms DCR 5MHz	0,81
L2	Bobina 22uH, 787 mOhms DCR 29.4 MHz	0,53
L3	Bobina 10uH 10%	0,53
L4, L7	Bobina toroidal 150UH 15% HORIZ	3,99
L5, L8	Núcleo de ferrita para fusibles 60 OHM 6000MA 1806	0,28
L6	Filtro de modo común 3.3uH, 2.5A	2,58
L9	Inductor fijo 0.22uH 40.5A SMD	2,49
MOV1, MOV2	Varistor 275V RMS 14mm Radial	0,96
Q1	Transistor bipolar NPN GP 500mA 45V SOT23	0,12
Q2, Q3, Q4, Q5	Transistor Mosfet Canal N 600V 20.2A TO263	2,2
Q6, Q8	Transistor Mosfet Canal P 150V 8.9A 1212-8	2,85
Q7, Q9	Transistor Mosfet Canal N 150V 83A D2PAK	7,59
R1, R10, R52, R57, R64, R69, R78, R80, R86, R88, R93, R94, R104, R107, R110, R111, R118, R119, R126, R127	Resistencia 10 KOhms 1/10W 1% 0603 SMD	0,02
R2, R32, R37, R38, R39, R41, R42, R44, R45	Resistencia 3.3 KOhms 1/10W 1%	0,015
R3	Resistencia 165 KOhms 1/10W 1% 0603 SMD	0,262
R4	Resistencia 10.2 KOhms 1/10W 1% 0603 SMD	0,014
R5	Resistencia 1.5 Ohms 1/4W 5% 1206 SMD	0,122
R6	Resistencia 169K Ohms 1/10W 1% 0603 SMD	0,013
R7	Resistencia 2.70K Ohms 1/10W 1% 0603 SMD	0,1
R8, R133, R134, R137, R138	Resistencia 0.0 Ohms 1/4W 1206 SMD	0,048
R9	Resistencia 52.3 KOhms 1/10W 1% 0603 SMD	0,306
R13, R22, R43, R47, R56, R59, R68, R70, R112, R117, R120, R125, R135, R136	Resistencia 0.0 Ohms 1/10W 0603 SMD	0,007

Nombre	Componentes	Precio (€/ud.)
R14	Resistencia 4.70 KOhms 1/10W 1% 0603 SMD	0,064
R15	Resistencia 470 Ohms 1/10W 1% 0603 SMD	0,018
R16, R17, R18, R71, R129, R130	Resistencia 330 Ohms 1/10W 1%	0,1
R19, R25, R28, R34	Resistencia, 100 KOhms 1/10W 1%	0,1
R20, R27	Resistencia 1.60 KOhms 1/10W 1% 0603 SMD	0,013
R21, R106	Resistencia 5.10 KOhms 1/10W 1% 0603 SMD	0,003
R23, R36	Resistencia 6.2 KOhms 1/10W 5% 0603 SMD	0,08
R24, R31	Resistencia 27 KOhms 1/10W 5% 0603 SMD	0,026
R26, R30, R128	Resistencia 1.69 KOhms 1/10W 1% 0603 SMD	0,011
R29, R73, R91, R92, R113, R115, R121, R123	Resistencia 1 KOhms 1/10W 1% 0603 SMD	0,007
R35	Resistencia 2.4 KOhms 1/10W 5% 0603 SMD	0,064
R40	Resistencia 470 KOhms 1/10W 1%	9,73
R46, R55, R58, R67, R74	Resistencia 7.5 KOhms 1/10W 1% 0603 SMD	0,018
R48, R50, R60, R62	Resistencia 3.01 KOhms 1/10W 1%	0,018
R49, R61	Resistencia 120 Ohms 1/10W 1%	0,014
R54, R66	Resistencia 15.0 Ohms 1/10W 1% 0603 SMD	0,012
R72	Resistencia 120 KOhms 1/10W 1% 0603 SMD	0,028
R75, R76, R83, R84, R95, R96, R114, R116, R122, R124	Resistencia 11.0 Ohms 1/10W 1% 0603 SMD	0,1
R77, R79, R85, R87	Resistencia 1.0 Ohms 1/10W 5% 0603 SMD	0,003
R81, R82, R89, R90	Resistencia 270 Ohms 1/10W 1% 0603 SMD	0,028
R97, R99, R101	Resistencia 150 KOhms 1/4W 5% 1206 SMD	0,022
R98, R100	Resistencia 240 KOhms 1/4W 5% 1206 SMD	0,118
R102	Resistencia 300 KOhms 1/4W 5% 1206 SMD	0,044
R103	Resistencia 100 Ohms 1/10W 5% 0603 SMD	0,055
R105	Resistencia 30 KOhms 1/10W 5% 0603 SMD	0,005
R108	Resistencia 20.0 KOhms 1/10W 1% 0603 SMD	0,003
R109	Resistencia 15.0 KOhms 1/10W 1% 0603 SMD	0,003
R141	Resistencia 3.3 KOhms 1/10W 1% 0603 SMD	0,015
SW1	Interruptor táctil SPST-NO 0.05A 12V	0,184
TP1, TP2, TP3, TP4, TP5, TP6, TP7, TP8, TP9, TP10, TP11, TP12, TP13, TP14, TP15, TP16, TP17, TP18, TP19, TP20, TP21, TP22, TP23, TP24, TP25, TP26, TP27, TP28, TP29, TP30, TP31	Utillaje puntos de medida	0,555
TR1	Transformador 1VA, 115/230V 10VCT 5V 10VCT 90mA	11,1
TR2, TR3	Transformador 264uH 1.5 DCR 3.6MH, 95pF	1,43
TR4, TR6	Núcleo de transformador RM14 230Vac	10,31
Base de TR4 y TR6	Soporte núcleo bobina Coilformer RM14 12-pins	1,368

Nombre	Componentes	Precio (€/ud.)
TR5, TR7	Transformador del sensor de corriente3000uH SMD	3,18
U1	Microprocesador High-Performance 32-bit Digital Signal Controller, 44-Pin QFN	5,05
U2	Regulador de conmutación IC BUCK ADJ .35A 8MSOP	2,72
U3	Convertidor de tensión IC REG BUCK ADJ 0.6A SOT23-6	0,943
U4	Regulador de tensión 3.3V 250MA SOT-89-3	0,38
U5, U6, U7	Amplificador operacional dual 2.5V 8SOIC	1,28
U8	Sensor de temperature lineal SC70-5	0,23
U9, U10, U11, U12	Optoacoplador para salida de transistores 4-SMD	0,456
U13, U16, U17	Driver IC MOSFET DVR 4.0A DUAL 8SOIC	1,77
U14	Sensor de corriente de efecto hall 5A 5V 8-SOIC	3,85
U15	Amplificador de error con aislamiento óptico 8-SMD	0,6
SW2	Interruptor ON/OFF 10A 250V	1,88
J3	Conector hembra para PCB .100"	0,65
CIR	Circuito impreso	70.48

2.2. Equipos y herramientas

Elemento	Precio (€/ud.)
Ordenador	798,5
Programador MPLAB ICD3	154,6
Fuente alimentación 30V 5A	219,9
Osciloscopio 4 canales	1580
Polímetro	45
Destornillador, alicates y otras herramientas	80
Regleta de protección	15,3
Pinza amperimétrica	330

2.3. Software

Programa	Precio (€/ud.)
Microsoft Word	135
Microsoft Excel	135
Eagle Profesional 6.1	Versión libre distribución
MPLAB v8.85	Versión libre distribución

2.4. Mano de obra directa

Actividad	Precio (€/hora)
Diseño de tarjeta	40
Búsqueda de componentes	25
Programación	40
Pruebas y solución de problemas	60
Documentación del proyecto	40

2.5. Servicios Auxiliares

Actividad	Precio (€)
Proceso de fabricación	300,00

Capítulo 3

Sumas parciales

En este capítulo se consignan los importes parciales de cada uno de los elementos que conforman las partidas, que han sido calculadas a partir de las mediciones y precios unitarios.

3.1. Componentes principales

Nombre	Componentes	Uds	Precio (€/ud.)	Pedido mínimo	Coste total (€)
C1	Condensador capacitivo cerámico 1.0UF, 100V X7R 1206	1	0,282	10	2,82
C2	Condensador capacitivo cerámico 10uF, 100V X7R 10% 0805	1	0,07	100	7,00
C3, C32, C44, C45, C46, C47, C80, C81	Condensador capacitivo cerámico 0.01uF, 50V X7R	8	0,006	1	0,05
C4, C7, C8, C9, C11, C14, C21, C22, C23, C24, C28, C35, C40, C54, C57, C74, C75, C77, C78, C82, C89	Condensador capacitivo cerámico 0.1uF, 50V X7R	21	0,014	1	0,29
C5, C20, C25, C29, C39, C55, C76, C79	Condensador capacitivo cerámico 1uF, 16V X5R	8	0,062	50	3,10
C6	Condensador electrolítico de Tántalo 100uF, 16V 10% 2917	1	1,004	5	5,02
C10, C13, C16, C17, C18, C19	Condensador capacitivo cerámico 10uF, 16V X5R	6	0,21	150	31,50
C27, C33, C37	Condensador capacitivo cerámico 8200pF, 10% 50V X7R 0603	3	0,096	10	0,96
C30	Condensador cerámico multicapa 2200pF, 0603 50V X7R 20%	1	0,444	25	11,10
C41, C42, C43	Condensador cerámico multicapa 470pF, 50V CERAMIC X7R 0603	3	0,033	10	0,33
C48, C49, C52, C53	Condensador capacitivo Cerámico 4700pF, 250VAC X1Y2 Rad	4	0,592	1	2,37
C87	Condensador capacitivo Cerámico 4700pF, 250VAC X1Y2 Rad	1	0,592	1	0,59
C50, C51, C88	Condensador de película MMKP 0.015uF, 1000VDC 5%	3	0,76	1	2,28
C56	Condensador cerámico multicapa 1000pF, 50V X7R 0603	1	0,192	1	0,19
C58, C69	Condensador de película 0.1uF, 630VDC RADIAL	2	0,337	10	3,37
C59, C70	Condensador capacitivo Cerámico 2200pF, 5% 100V NP0 1210	2	0,568	1	1,14

Nombre	Componentes	Uds	Precio (€/ud.)	Pedido mínimo	Coste total (€)
C60, C71	Condensador capacitivo Cerámico 10uF, 100V X7S 2220	2	1,766	5	8,83
C61, C62, C72, C73, C83, C84, C85, C86	Condensador capacitivo Cerámico 2.2uF, 250V 20% X7T 2225	8	1,73	10	17,30
C63, C64, C65, C66, C67	Condensador electrolítico de Aluminio 2200uF, 63V 20% Radial	5	2,3	2	11,50
C68	Condensador capacitivo Cerámico 120pF, 50V 5% C0G 0603	1	0,071	50	3,55
C90	Condensador capacitivo Cerámico 220pF, 50V 10% X7R 0603	1	0,039	10	0,39
D1	Diodo Schottky 1A, 60V SMA	1	0,104	1	0,10
D2	Diodo switch 75V 300mA, SOD323	1	0,163	1	0,16
D3, D9, D11	Diodo Schottky 500mA, 20V 0603	3	0,352	1	1,06
D4, D26, D27	LED Rojo, SMD 0603	3	0,104	1	0,31
D5	LED Verde, SMD 0603	1	0,112	1	0,11
D6	LED Amarillo SMD 0603	1	0,112	1	0,11
D7, D8, D10, D12, D28	Diodo Schottky 200mA, 30V SOT23	5	0,143	10	1,43
D13	Referencia tensión 2,5V 1% Aj. SOT23F	1	0,124	25	3,10
D14, D15, D16, D17, D20, D21, D22, D23, D24, D25	Diodo Schottky 500MA, 40V SOD123	10	0,192	10	1,92
D18, D19	Diodo Schottky 10A, 1200V TO252-2	2	6,4	1	12,80
D29	Diodo Zener TVS Bidireccional 400W 10V SMA	1	0,78	1	0,78
F1, F2	Fusible Lento 250VAC 6.3A Radial	2	0,663	10	6,63
J1	Conector modular Jack 6-6 RT/A PCB 50AU	1	0,97	5	4,85
J2	Conector Terminal Strip, 2X6, 0.100sp, 0.025 sq post	1	1,48	10	14,80
J3	Conector Terminal Strip, 1X2, 0.100sp, 0.025 sq post	1	0,325	10	3,25
J4	Conector con barreras Strip 3POS .375"	1	3,18	1	3,18
J5, J6	Conector PCB .250	2	0,068	1	0,14
L1	Bobina 220uH, 700 mOhms DCR 5MHz	1	0,81	1	0,81
L2	Bobina 22uH, 787 mOhms DCR 29.4 MHz	1	0,53	1	0,53
L3	Bobina 10uH 10%	1	0,53	1	0,53
L4, L7	Bobina toroidal 150UH 15% HORIZ	2	3,99	1	7,98
L5, L8	Núcleo de ferrita para fusibles 60 OHM 6000MA 1806	2	0,28	10	2,80
L6	Filtro de modo común 3.3uH, 2.5A	1	2,58	1	2,58
L9	Inductor fijo 0.22uH 40.5A SMD	1	2,49	1	2,49

Nombre	Componentes	Uds	Precio (€/ud.)	Pedido mínimo	Coste total (€)
MOV1, MOV2	Varistor 275V RMS 14mm Radial	2	0,96	5	4,80
Q1	Transistor bipolar NPN GP 500mA 45V SOT23	1	0,12	1	0,12
Q2, Q3, Q4, Q5	Transistor Mosfet Canal N 600V 20.2A TO263	4	2,2	1	8,80
Q6, Q8	Transistor Mosfet Canal P 150V 8.9A 1212-8	2	2,85	1	5,70
Q7, Q9	Transistor Mosfet Canal N 150V 83A D2PAK	2	7,59	1	15,18
R1, R10, R52, R57, R64, R69, R78, R80, R86, R88, R93, R94, R104, R107, R110, R111, R118, R119, R126, R127	Resistencia 10 KOhms 1/10W 1% 0603 SMD	20	0,02	50	1,00
R2, R32, R37, R38, R39, R41, R42, R44, R45	Resistencia 3.3 KOhms 1/10W 1%	9	0,015	50	0,75
R3	Resistencia 165 KOhms 1/10W 1% 0603 SMD	1	0,262	5	1,31
R4	Resistencia 10.2 KOhms 1/10W 1% 0603 SMD	1	0,014	50	0,70
R5	Resistencia 1.5 Ohms 1/4W 5% 1206 SMD	1	0,122	5	0,61
R6	Resistencia 169K Ohms 1/10W 1% 0603 SMD	1	0,013	50	0,65
R7	Resistencia 2.70K Ohms 1/10W 1% 0603 SMD	1	0,1	5	0,50
R8, R133, R134, R137, R138	Resistencia 0.0 Ohms 1/4W 1206 SMD	5	0,048	50	2,40
R9	Resistencia 52.3 KOhms 1/10W 1% 0603 SMD	1	0,306	5	1,53
R13, R22, R43, R47, R56, R59, R68, R70, R112, R117, R120, R125, R135, R136	Resistencia 0.0 Ohms 1/10W 0603 SMD	14	0,007	50	0,35
R14	Resistencia 4.70 KOhms 1/10W 1% 0603 SMD	1	0,064	1	0,06
R15	Resistencia 470 Ohms 1/10W 1% 0603 SMD	1	0,018	50	0,90
R16, R17, R18, R71, R129, R130	Resistencia 330 Ohms 1/10W 1%	6	0,1	5	0,60
R19, R25, R28, R34	Resistencia, 100 KOhms 1/10W 1%	4	0,1	5	0,50
R20, R27	Resistencia 1.60 KOhms 1/10W 1% 0603 SMD	2	0,013	50	0,65
R21, R106	Resistencia 5.10 KOhms 1/10W 1% 0603 SMD	2	0,003	1	0,01
R23, R36	Resistencia 6.2 KOhms 1/10W 5% 0603 SMD	2	0,08	1	0,16
R24, R31	Resistencia 27 KOhms 1/10W 5% 0603 SMD	2	0,026	50	1,30
R26, R30, R128	Resistencia 1.69 KOhms 1/10W 1% 0603 SMD	3	0,011	50	0,55
R29, R73, R91, R92, R113, R115, R121, R123	Resistencia 1 KOhms 1/10W 1% 0603 SMD	8	0,007	50	0,35

Nombre	Componentes	Uds	Precio (€/ud.)	Pedido mínimo	Coste total (€)
R35	Resistencia 2.4 KOhms 1/10W 5% 0603 SMD	1	0,064	1	0,06
R40	Resistencia 470 KOhms 1/10W 1%	1	9,73	1	9,73
R46, R55, R58, R67, R74	Resistencia 7.5 KOhms 1/10W 1% 0603 SMD	5	0,018	50	0,90
R48, R50, R60, R62	Resistencia 3.01 KOhms 1/10W 1%	4	0,018	50	0,90
R49, R61	Resistencia 120 Ohms 1/10W 1%	2	0,014	50	0,70
R54, R66	Resistencia 15.0 Ohms 1/10W 1% 0603 SMD	2	0,012	50	0,60
R72	Resistencia 120 KOhms 1/10W 1% 0603 SMD	1	0,028	50	1,40
R75, R76, R83, R84, R95, R96, R114, R116, R122, R124	Resistencia 11.0 Ohms 1/10W 1% 0603 SMD	10	0,1	5	1,00
R77, R79, R85, R87	Resistencia 1.0 Ohms 1/10W 5% 0603 SMD	4	0,003	1	0,01
R81, R82, R89, R90	Resistencia 270 Ohms 1/10W 1% 0603 SMD	4	0,028	50	1,40
R97, R99, R101	Resistencia 150 KOhms 1/4W 5% 1206 SMD	3	0,022	50	1,10
R98, R100	Resistencia 240 KOhms 1/4W 5% 1206 SMD	2	0,118	5	0,59
R102	Resistencia 300 KOhms 1/4W 5% 1206 SMD	1	0,044	50	2,20
R103	Resistencia 100 Ohms 1/10W 5% 0603 SMD	1	0,055	50	2,75
R105	Resistencia 30 KOhms 1/10W 5% 0603 SMD	1	0,005	1	0,01
R108	Resistencia 20.0 KOhms 1/10W 1% 0603 SMD	1	0,003	1	0,00
R109	Resistencia 15.0 KOhms 1/10W 1% 0603 SMD	1	0,003	1	0,00
R141	Resistencia 3.3 KOhms 1/10W 1% 0603 SMD	1	0,015	50	0,75
SW1	Interruptor táctil SPST-NO 0.05A 12V	1	0,184	1	0,18
TP1, TP2, TP3, TP4, TP5, TP6, TP7, TP8, TP9, TP10, TP11, TP12, TP13, TP14, TP15, TP16, TP17, TP18, TP19, TP20, TP21, TP22, TP23, TP24, TP25, TP26, TP27, TP28, TP29, TP30, TP31	Utillaje puntos de medida	31	0,555	1	17,21
TR1	Transformador 1VA, 115/230V 10VCT 5V 10VCT 90mA	1	11,1	1	11,10
TR2, TR3	Transformador 264uH 1.5 DCR 3.6MH, 95pF	2	1,43	1	2,86
TR4, TR6	Núcleo de transformador RM14 230Vac	2	10,31	1	20,62
Base de TR4 y TR6	Soporte núcleo bobina Coilformer RM14 12-pins	2	1,368	5	6,84
TR5, TR7	Transformador del sensor de corriente3000uH SMD	2	3,18	1	6,36
U1	Microprocesador High-Performance 32-bit Digital Signal Controller, 44-Pin QFN	1	5,05	1	5,05
U2	Regulador de conmutación IC BUCK ADJ .35A 8MSOP	1	2,72	1	2,72

Nombre	Componentes	Uds	Precio (€/ud.)	Pedido mínimo	Coste total (€)
U3	Convertidor de tensión IC REG BUCK ADJ 0.6A SOT23-6	1	0,943	4	3,77
U4	Regulador de tensión 3.3V 250MA SOT-89-3	1	0,38	1	0,38
U5, U6, U7	Amplificador operacional dual 2.5V 8SOIC	3	1,28	5	6,40
U8	Sensor de temperature lineal SC70-5	1	0,23	10	2,30
U9, U10, U11, U12	Optoacoplador para salida de transistores 4-SMD	4	0,456	1	1,82
U13, U16, U17	Driver IC MOSFET DVR 4.0A DUAL 8SOIC	3	1,77	2	5,31
U14	Sensor de corriente de efecto hall 5A 5V 8-SOIC	1	3,85	1	3,85
U15	Amplificador de error con aislamiento óptico 8-SMD	1	0,6	5	3,00
SW2	Interruptor ON/OFF 10A 250V	1	1,88	1	1,88
J3	Conector hembra para PCB .100"	1	0,65	1	0,65
CIR	Circuito impreso	1	70.48	1	40,48

3.2. Equipos y herramientas

Para calcular el coste imputable del equipo y las herramientas, se ha supuesto un tiempo de amortización típico de 4 años y se ha repartido de forma alícuota en función del número de horas empleadas.

Elemento	Uds	Horas de proyecto	Horas por año	Precio (€/ud.)	Amortización anual	Coste (€)
Ordenador	1	500	1600	798,5	25%	62,38
Programador MPLAB ICD3	1	200	500	154,6	25%	15,46
Fuente alimentación 30V 5A	1	200	400	219,9	25%	27,49
Osciloscopio 4 canales	1	20	300	1580	25%	26,33
Polímetro	1	50	250	45	25%	2,25
Destornillador, alicates y otras herramientas	1	20	400	80	25%	1,00
Regleta de protección	1	500	1600	15,3	25%	1,20
Pinza amperimétrica	1	20	300	330	25%	5,50
					Total	141,61

3.3. Software

Programa	Uds	Horas de proyecto	Horas por año	Precio (€/ud.)	Amortización anual	Coste (€)
Microsoft Word	1	100	500	135,00	20%	5,40
Microsoft Excel	1	150	500	135,00	20%	8,10
					Total	13,50

3.4. Mano de obra directa

Actividad	Horas	Precio (€/hora)	Coste total (€)
Diseño de tarjeta	100	40	4.000,00
Búsqueda de componentes	100	25	2.500,00
Programación	200	40	8.000,00
Pruebas y solución de problemas	150	60	9.000,00
Documentación del proyecto	100	40	4.000,00
		Total	27.500,00

3.5. Servicios Auxiliares

Actividad	Uds	Precio (€)	Coste total (€)
Proceso de fabricación	1	300,00	300,00

Capítulo 4

Presupuesto general

Sumando la contribución de todas las partidas anteriores se concluye que el coste del proyecto, impuestos incluidos, asciende a:

Concepto	Coste (€)
Componentes principales	432,50
Equipo y herramientas	141,61
Software	13,50
Mano de obra directa	27.500,00
Servicios auxiliares	300,00
Total	28.387,61